The Journal of Physical Chemistry Letters

Letter

Cite This: J. Phys. Chem. Lett. 2019, 10, 1242−1250

pubs.acs.org/JPCL

Downloaded via UNIV ILLINOIS URBANA-CHAMPAIGN on March 13, 2019 at 15:50:18 (UTC).
See https://pubs.acs.org/sharingguidelines for options on how to legitimately share published articles.

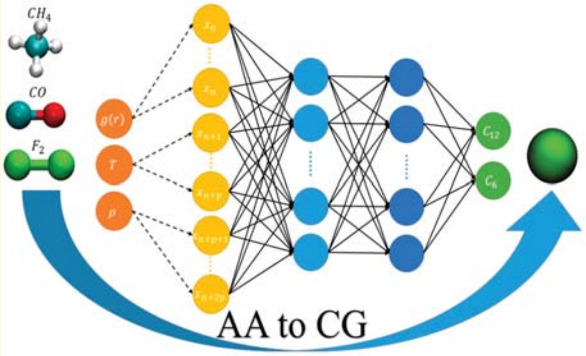
Transfer-Learning-Based Coarse-Graining Method for Simple Fluids: Toward Deep Inverse Liquid-State Theory

Alireza Moradzadeh and Narayana R. Aluru*

Department of Mechanical Science and Engineering, Beckman Institute for Advanced Science and Technology, University of Illinois at Urbana−Champaign, Urbana, Illinois 61801, United States

Supporting Information

ABSTRACT: Machine learning is an attractive paradigm to circumvent difficulties associated with the development and optimization of force-field parameters. In this study, a deep neural network (DNN) is used to study the inverse problem of the liquid-state theory, in particular, to obtain the relation between the radial distribution function (RDF) and the Lennard-Jones (LJ) potential parameters at various thermodynamic states. Using molecular dynamics (MD), many observables, including RDF, are determined once the interatomic potential is specified. However, the inverse problem (parametrization of the potential for a specific RDF) is not straightforward. Here we present a framework integrating DNN with big data from 1.5 TB of MD trajectories with a cumulative simulation time of 52 μs for 26 000 distinct systems to predict LJ potential parameters. Our results show that DNN is successful not only in the parametrization of the atomic LJ liquids but also in parametrizing the LJ potential for coarse-grained models of simple multiatom molecules.



The Lennard-Jones (LJ) potential form is one of the widely used nonbonded 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).[1] The 12−6 LJ potential is prominently used in molecular dynamics (MD) simulations to study physical, chemical, biological, and mechanical systems.[2] 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.[3] As per Henderson's theorem,[4] 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 Henderson's theorem using a data-driven approach combined with deep learning (DL).

The inverse problem of parametrization 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.[5−8] Different frameworks, such as the fundamental measure theory[9,10] and integral equations,[11] 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[5,12]) or used to optimize statistical or empirical errors between the MD-calculated and given RDFs (e.g., relative entropy or the simplex minimization method[13,14]). MD-based approaches require thousands of simulations on a specific system of interest, and the data are often not reused for other systems with similar mapping and underlying potential form (see Figure 1a).[15] The main bottleneck in reusing data for the parametrization 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 obtain the underlying physics of data, also known as the physics of big data,[16] by surrogate models. Surrogate models circumvent the difficulty associated with the 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 closed-form analytical solution or repetitive MD simulation of a specific system.

Received: December 30, 2018
Accepted: February 28, 2019
Published: February 28, 2019

ACS Publications © XXXX American Chemical Society

DOI: 10.1021/acs.jpclett.8b03872

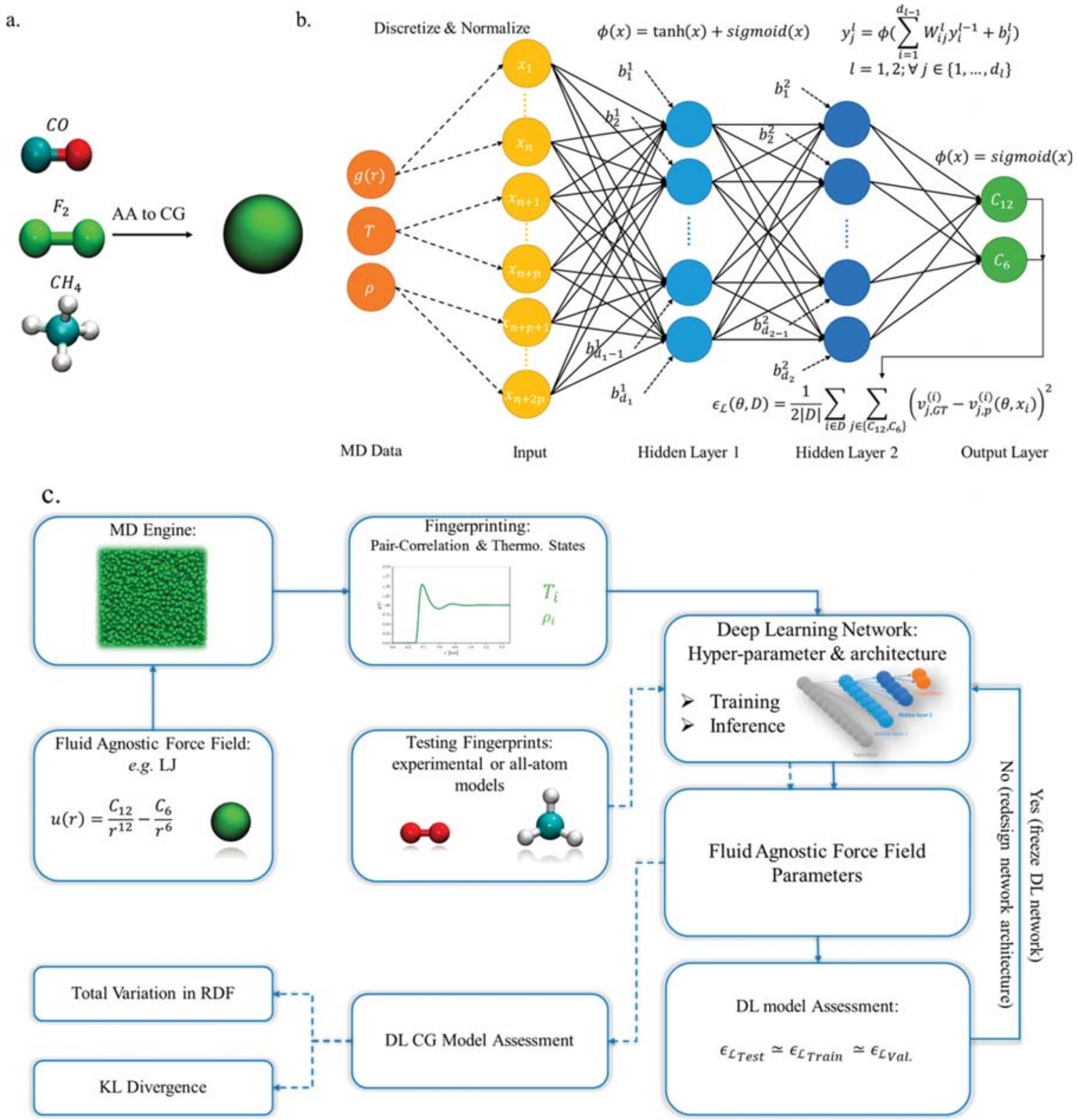

a.
CO
F2
CH4
AA to CG
b.
Discretize & Normalize
ϕ(x) = tanh(x) + sigmoid(x)
y_j^l = ϕ(Σ W_ij^l y_i^{l−1} + b_j^l)
l = 1, 2; ∀ j ∈ {1, …, d_l}
g(r)
T
ρ
x1
xn
xn+1
xn+p
xn+p+1
xn+2p
ϕ(x) = sigmoid(x)
C12
C6
ε_L(θ, D) = 1/(2|D|) Σ_{i∈D} Σ_{j∈{C12,C6}} (v_{j,GT}^{(i)} − v_{j,p}^{(i)}(θ, x_i))^2
MD Data
Input
Hidden Layer 1
Hidden Layer 2
Output Layer
c.
MD Engine:
Fingerprinting:
Pair-Correlation & Thermo. States
T_i
ρ_i
Deep Learning Network:
Hyper-parameter & architecture
➢ Training
➢ Inference
Fluid Agnostic Force Field:
e.g. LJ
u(r) = C12/r^12 − C6/r^6
Testing Fingerprints:
experimental or all-atom models
Fluid Agnostic Force Field Parameters
No (redesign network architecture)
Yes (freeze DL network)
Total Variation in RDF
KL Divergence
DL CG Model Assessment
DL model Assessment:
ε_L_Test ≃ ε_L_Train ≃ ε_L_Val.

Figure 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% 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% 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 are accomplished, either RDFs obtained from AA models or experimental data are used as input to the DNN to predict the force-field parameters. (Transfer-learning-based 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.

During the past decade, data-driven and machine-learning (ML) methods have received enormous attention in the computational analysis of physical problems.[16−19] Even though the usage of ML methods in scientific research and discovery dates back to several decades ago, only in the last 5 years researchers have embarked on using DL[20] in the modeling and understanding of physical phenomena such as data-driven materials discovery,[21−26] calculation of the thermodynamic and ground states in multibody systems,[27,28] phase transitions,[29] classification of phases in strongly correlated Fermions,[30] quantum entanglement,[31] and many other applications.[32] During the last couple of years, researchers have started to develop deep neural networks (DNNs) for CG model development.[33,34] 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 the RDFs of LJ particles at various thermodynamic states with the potential parameters (C_{12} and C_6). The data set 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 multiatom molecules, with bonded and nonbonded 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 the 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 (KL) divergence. 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 multiatom 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 paper is organized as follows. First, we describe the details of MD simulations for both multiatom and single-bead molecules and DNN development as well as coarse-graining through transfer learning. Then, we present the results of DL training as well as coarse-graining through transfer learning. Finally, we summarize the findings of this study.

The data for DNN training are generated via MD simulation of 26 000 distinct systems with interaction potential parameters and thermodynamic states sampled uniformly over the range shown in Table 1. The LJ potential form is given by

$$u(r) = \frac{C_{12}}{r^{12}} - \frac{C_6}{r^6} \tag{1}$$

Table 1. Thermodynamic States and the Range of Parameters Used during MD Simulation of Single-Bead Systems

	thermodynamic state		parameter range	
	ρ (nm^{-3})	T (K)	$C_{12}\left[\frac{\text{kJ}}{\text{mol}\cdot\text{nm}^{12}}\right]$	$C_6\left[\frac{\text{kJ}}{\text{mol}\cdot\text{nm}^{6}}\right]$
min	8.0	290.0	0.000005	0.00005
max	19.4	400.0	0.0001	0.01

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 of the MD simulations are performed using GROMACS[35] with a time step 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 and its corresponding thermodynamic state 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 multiatom molecules such as CH_4, F_2, and CO are simulated to evaluate the development of CG potentials for these multiatom molecules using the DNN developed for LJ particles. The simple multiatom molecules, shown in Figure 1a, are modeled using the GROMOS force field.[36] The interactions between simple multiatom molecules are described by both bonded and nonbonded potentials. The bonded interactions are modeled according to GROMOS force-field bond and angle potential form.[36] 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, $u_{\text{Coulomb}}(\boldsymbol{r}) = \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}}$, where ϵ_0 denotes the dielectric permittivity of the 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.[37] 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.

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

$$(C_{12}, C_6) = \boldsymbol{f}(g(r); T, \rho) \tag{2}$$

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,$g(r) = \exp\left(-\frac{U_{\text{PMF}}(r)}{k_B T}\right)$, 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 based on the universal approximation theorem, which states that FNN with enough capacity can approximate many continuous functions.[38] The data of each MD simulation are fed into DNN as

$$x_i = (\boldsymbol{g}_i(r), \rho_i^1, \rho_i^2, ..., \rho_i^p, T_i, T_i^2, ..., T_i^p)_m \tag{3}$$

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 data set (D) with a total size of m ($= n + 2p$) and i refers to the ith LJ system ($x_i \in D$). The first n elements correspond to the value of RDF discretized between its minimum and maximum range, that is, ($x_{i,l} = g_i(r_l < r \leq r_l + dr)/\max_{i \in D} g_i(r)\ \forall\ l \in \{1, 2, ..., n\}$, 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; for example, $x_{i,n+l} = (\rho_i^l - \min_{i \in D} \rho_i^l)/(\max_{i \in D} \rho_i^l - \min_{i \in D} \rho_i^l)\ \forall\ l \in \{1, 2, ..., p\}$ defines 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 (stacking layers successively; hidden layers are shown with blue circles in Figure 1b) results in a multilayer perceptron, which can approximate many continuous functions.[20,39] With x_i as the input and the LJ potential parameters as the output, the DNN can be mathematically expressed as

$$(C_{12,i}^{\text{fnn}}, C_{6,i}^{\text{fnn}}) = \phi_o(\boldsymbol{W}_o \phi_{n_h}(... \phi_2(\boldsymbol{W}_2 \phi_1(\boldsymbol{W}_1 x_i + \boldsymbol{b}_1) + \boldsymbol{b}_2)) + \boldsymbol{b}_o) \tag{4}$$

where ϕ_k is the nonlinear activation function of layer k given by $\phi_k(x) = \phi_k(\boldsymbol{W}_k\phi_{k-1}(x) + \boldsymbol{b}_k) = \tanh(\boldsymbol{W}_k\phi_{k-1}(x) + \boldsymbol{b}_k) + \mathrm{sigmoid}(\boldsymbol{W}_k\phi_{k-1}(x) + \boldsymbol{b}_k)$. 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. The summation of the tangent hyperbolic and sigmoid nonlinearities ($\tanh x$ + sigmoid x) is used between layers, except for the output layer (ϕ_o) where the sigmoid nonlinearity is used (see eqs S4−S7).

As shown in Figure 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[40] is used to minimize the loss function ($\epsilon_\mathcal{L}(\theta, D)$), which can be expressed as

$$\epsilon_\mathcal{L}(\theta, D) = \frac{1}{2|D|}\sum_{i\in D}\ \sum_{j\in\{C_{12},C_6\}} (v_{j,\mathrm{GT}}^{(i)} - v_{j,p}^{(i)}(\theta, x_i))^2 \tag{5}$$

where θ represents the free parameters (weights and biases) of DNN. $v_{j,\mathrm{GT}}^{(i)}$ 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 ith data point in the data set. $|D|$ denotes the size of the training data set. The training data set corresponds to ∼75% of the data obtained from MD simulations, and it are used to find the optimal weights and biases of the DNN. The validation data set (∼12.5% of the MD data set) is used as a metric during the loss function minimization to monitor network performance, avoiding overfitting or underfitting in the learning process; that is, it is fed into the network to determine the value of the loss function, but it does not contribute to the backpropagation. Furthermore, during training, we use the dropout technique for the second hidden layer to avoid overfitting.[41] 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 coadapting, and they far outweigh the ones achieved with L1 and L2 regularizations.

The training and design of DNN continue until a good performance is achieved on both the training and validation data sets. Once the network is trained, we check its loss function value on the testing data set (the remaining 12.5% of MD data set) to ensure its generalizability. If the network performance shows overfitting or underfitting on the validation or training data sets, then its architecture involving the number of layers, the number of nodes in each layer, and the activation functions are changed by trying rectified linear unit, sigmoid, tangent hyperbolic, or a combination of them until a reasonable performance is achieved. On the basis of the network performance on the entire data set, 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 of the results presented in the main manuscript are obtained using this network. (See the Supporting Information Section S.2 for more details.)

Next, we consider coarse-graining simple multiatom molecules into single beads to preserve the structure (RDF) of the multiatom molecule. Specifically, a multiatom molecule is coarse-grained into a single bead such that the COM RDF of the multiatom 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 1c (dashed arrows in the figure), the COM RDF and the thermodynamic state of a simple multiatom 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 multiatom molecule. Once the potential parameters are available, MD simulations can be performed using the single-bead representation of the multiatom molecule to estimate RDF and other properties of the CG multiatom system. It is important to note that, typically, coarse-graining introduces errors in the estimation of various properties of the original multiatom 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.[13,15] The error in the total variation between CG and AA RDFs is calculated using the expression

$$\epsilon_\mathrm{rdf} = \frac{\int_0^{r_{cf}} |g_\mathrm{CG}(r) - g_\mathrm{AA}(r)| r^2\, \mathrm{d}r}{\int_0^{r_{cf}} g_\mathrm{AA}(r) r^2\, \mathrm{d}r} \tag{7}$$

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

$$\epsilon_\mathrm{KL} = \beta\langle U_\mathrm{CG} - U_\mathrm{AA}\rangle_\mathrm{AA} - \beta\langle F_\mathrm{CG} - F_\mathrm{AA}\rangle_\mathrm{AA} + \langle S_\mathrm{map}\rangle_\mathrm{AA} \tag{8}$$

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 $\langle S_\mathrm{map}\rangle_\mathrm{AA}$ is the mapping entropy, which is not a function of the potential parameters. Appendix I contains mathematical details of KL divergence calculation as well as its convexity and the existence and uniqueness of a global minimum.[42,43]

We examine the performance of DNN by considering two cases: First, the generalizability and transferability of the interatomic potential parametrization 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 data set; that is, the thermodynamic state considered to establish generalizability is not part of the training data set, but the thermodynamic state lies within the training data set thermodynamic states that are part of the training data set. 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 data set; that is, the thermodynamic states considered are not within the range shown in Table 1. Transfer learning refers to the use of DNN to estimate potential parameters for CG representation of simple multiatom molecules shown in Figure 1a.

DNN is trained to reproduce the potential parameters of LJ particles using RDFs and thermodynamic-state variables. The

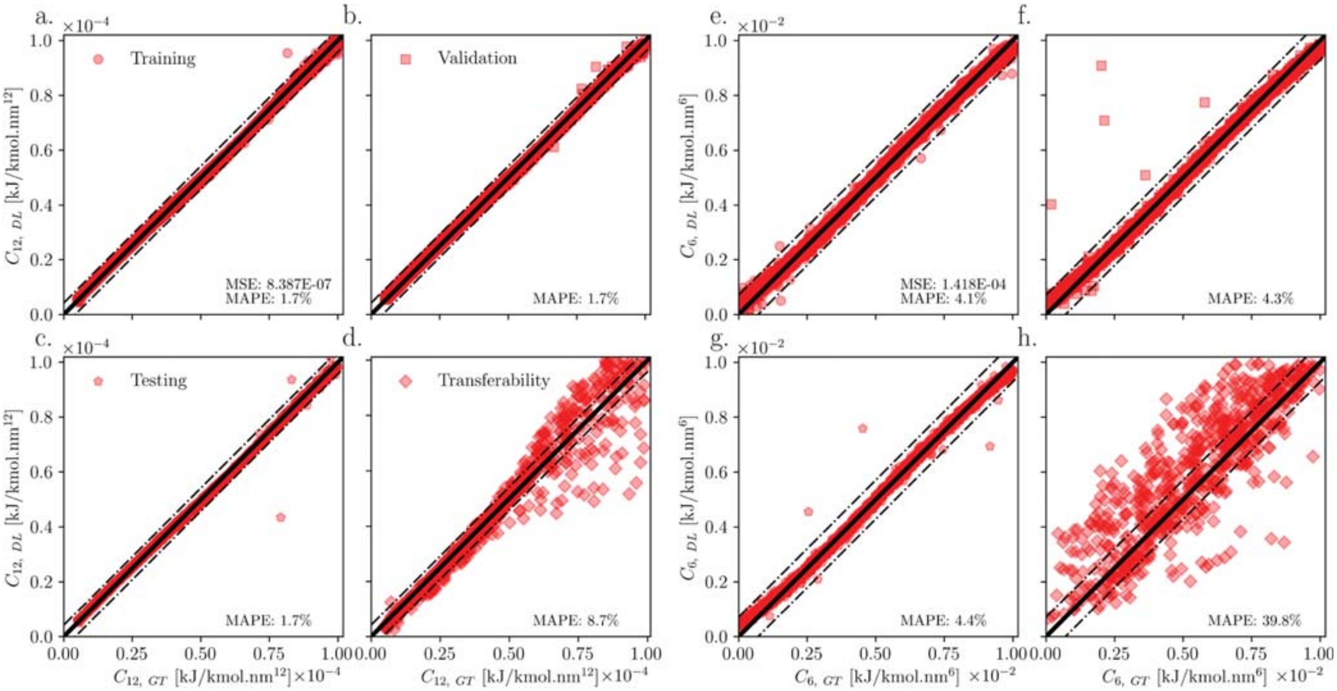


Figure 2. Comparison of the pair potential parameters determined from the DNN with the ground-truth values for training, validation, testing, and transferability data sets. (a−d) C_{12} parameter from training, validation, testing and transferability data sets. (e−h) C_6 parameter from training, validation, testing, and transferability data sets. The mean absolute percentage error is also shown for each data set. (Red circles, square, pentagon, and diamond points represent training, validation, testing, and transferability data sets, respectively, while black solid lines show a one-to-one mapping and black dashed-dotted 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 data set. 99% of the training data are enclosed in this region. The region also encloses ∼99% of the data of validation and testing data sets.)

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 data sets. (See Figure S4 for the loss function minimization during each iteration of training; the total number of iterations for each network is about 20 000 000.) The generalizability of the network is assessed through its performance on the test data set, 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. We note that all of the points are distributed almost uniformly around the one-to-one mapping line, that is, the line on which the ground-truth and DNN predicted parameters match exactly. Figure 2a,e shows the training data set results for the prediction of C_{12} and C_6, respectively. Whereas the thermodynamic states and RDF vary for each point, the DNN is able to relate them correctly to the underlying potential parameters. Similarly, Figure 2b,f shows the validation data set results for the predictability of C_{12} and C_6, respectively. The validation data set is used for the design of DNN to avoid overfitting and is not part of the data set used for the minimization of the loss function. Figure 2c,g shows 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 data set. (See eqs S8−S10 for the definition of bandwidth and accuracy as well as the definition of the mean absolute percentage error (MAPE),[44] defined as $\epsilon_{\mathrm{MAPE},j} = 100 \times \frac{\sum_{i\in D}|v_{j,\mathrm{DNN}}^{(i)} - v_{j,\mathrm{GT}}^{(i)}|}{\sum_{i\in \mathrm{D}}|v_{j,\mathrm{GT}}^{(i)}|}$, where i represents the ith data point in the data set and v_j is either C_{12} or C_6.) The dashed-dotted lines, shown in Figure 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 data set. These lines show the interval in which ∼99% of DNN prediction points are located compared with ground truth of the training data set. The same accuracy lines are shown for the rest of the data sets, which shows that most of the data of validation and testing data sets lie in this region (close to 99% of points). The MAPE is also shown for each data set and potential parameters. C_{12} and C_6 of training, validation, and testing data sets 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 data set 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 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 data set is shown in Figure 2d,h, indicating about 8.7 and 39.8% MAPE for C_{12} and C_6, respectively. Similarly, about half of the points of the transferability data set 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 a clear correlation between the predicted and ground-truth parameters. (See Supporting Information S.2 for a discussion on the details of the transferability data set 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 of the temperature and density ranges shown in Table 1, with each having 11 points. The pressure calculated using the DNN-estimated potential

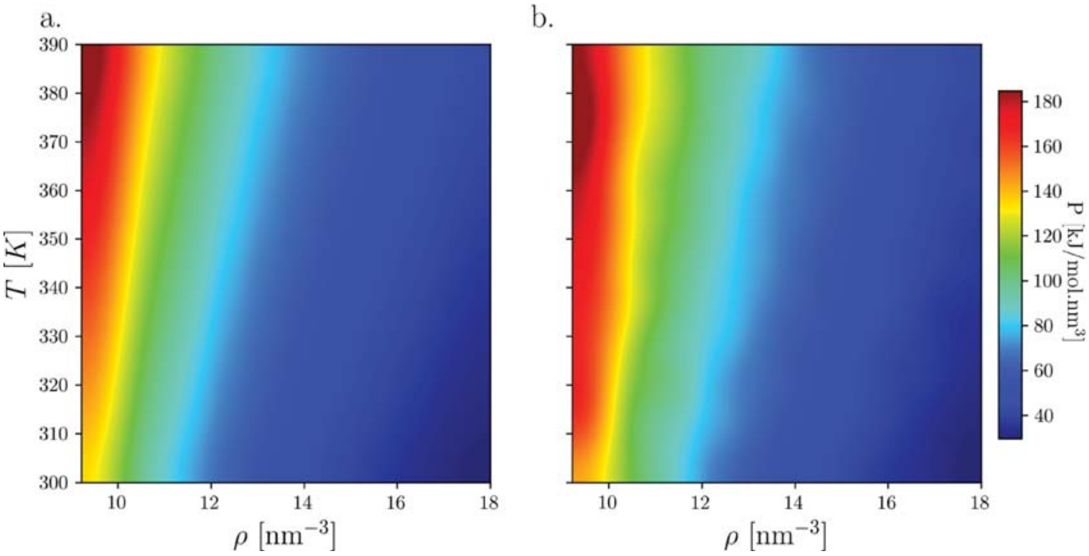


Figure 3. Calculation of the bulk pressure of argon for different thermodynamic states. (a) Molecular dynamics simulations. (b) Deep-learning-predicted potential parameter-based pressure.

parameters (shown in Figure 3) has a mean error of 17% compared with the AA MD pressure for the 121 points used in the data set. This error is reasonably small compared with 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. (The details of the calculation of the thermodynamic quantities including pressure and total energy are presented in Supporting Information S.1.) As Table 2 indicates, the network is able to map all 121 points into a single value of LJ parameter with a small deviation as well as a small error compared with the MD ground-truth parameters (6.18% for C_{12} and 0.16% for C_6 relative to the ground-truth LJ parameters of argon).

Table 2. Predictability of Argon Parameters on a 11 × 11 Mesh Covering Various Temperature and Density Values[a]

	ground truth	deep learning	relative error %
$C_{12} \times 10^{-6}\ [\frac{\text{kJ}}{\text{mol}\cdot\text{nm}^{12}}]$	9.70	10.3 ± 0.116	6.18
$C_6 \times 10^{-3}\ [\frac{\text{kJ}}{\text{mol}\cdot\text{nm}^{6}}]$	6.22	6.23 ± 0.0053	0.16

[a] MD simulations are performed on argon with ground-truth LJ potential parameters at different thermodynamic states on a uniform grid over the temperature and density ranges stated in Table 1. All 121 RDFs and their corresponding thermodynamic states are fed into the DNN, which provides estimates of LJ potential parameters.

The DNN results for the parametrization of LJ particles exhibit no more than 4.4% MAPE over the data set, 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 multiatom molecules such as carbon monoxide, fluorine, and methane are developed. The COM RDFs of these systems are first obtained using all-atom molecular dynamics (AAMD) simulations. Then, the COM RDFs and thermodynamic states (midpoint of Table 1) are fed into the DNN. (The procedure is shown with a dashed line in Figure 1c.) The comparison of the COM RDFs of multiatom molecules and CG model RDFs is shown in Figure 4. Figure 4 also shows the results of CG RDFs of multiatom molecules for the LJ pair potential parametrized by simplex and relative entropy methods. The

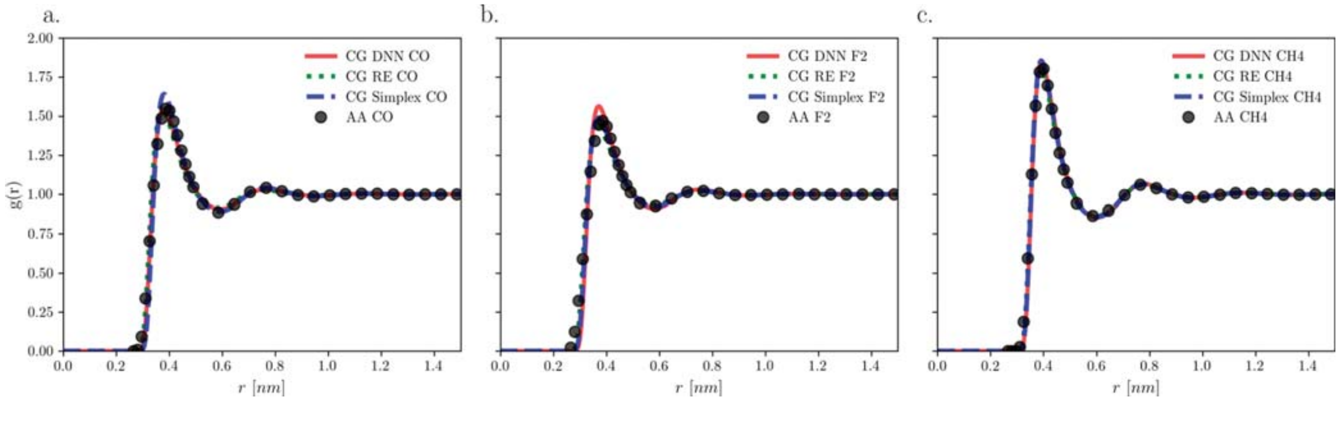


Figure 4. Comparison of RDFs obtained with DNN-based (solid red line), relative entropy (dotted green line), and simplex (dashed-dotted blue line) CG models and the AA model. All three methods show an excellent match with the AA model results (shown as black circles). (a) CO. (b) F_2. (c) CH_4.

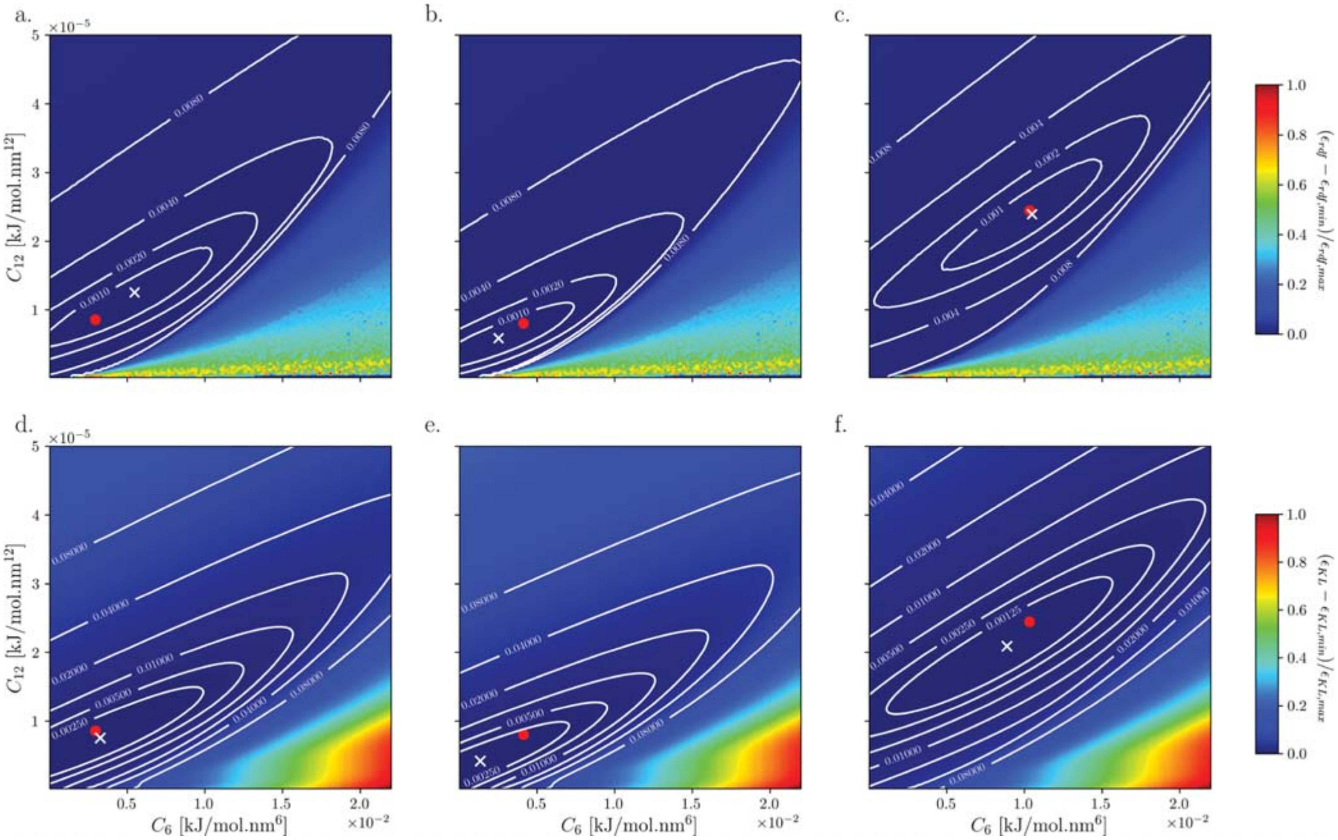


Figure 5. Assessment of the coarse-grained force fields through total variation in RDF (first row) and Kullback−Leibler divergence (second row) criteria. The parameter space is discretized into a mesh, and ϵ_{rdf} and ϵ_{KL} are computed at each mesh point. White cross points are the minimum values obtained from the two metrics, while red filled circles are predictions from deep learning. (a,d) CO. (b,e) F_2. (c,f) CH_4.

results indicate that DNN parametrized 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 with the simplex and relative entropy methods. (See Figures S11 and S12 as well as Table S1 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, that is, ϵ_{rdf} and ϵ_{KL}. In Figure 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 these 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. On the basis of eq 8, ϵ_{KL} between simple multiatom molecules and the LJ particles is shown in Figure 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 its convexity, see Appendix I or ref 13 and references therein.) Both error metrics show a small deviation from the DNN-predicted LJ parameters with a distance <0.1% of the maximum error in the investigated parameter space. Because 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 CG strategy because it performs well on both metrics.

In this study, a deep neural network is used for atom-agnostic parametrization of the LJ potential at different thermodynamic states. The DNN demonstrates good performance for two cases—parametrization of LJ particles and the development of single-bead CG LJ potentials for simple multiatom molecules through transfer learning obtained from LJ particles. The transferability and generalizability of the method are investigated by computing the total variation in the RDF and KL divergence for the CG model development. Our results indicate that DL 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 CG model.

■ APPENDIX I. KULLBACK−LEIBLER DIVERGENCE

The KL divergence or the relative entropy is a concept from information theory and mathematical statistics to measure the distance between two probabilities. It is defined as

$$\epsilon_{\text{KL}}(P;\, Q) = -\sum_i P(i)\log\frac{Q(i)}{P(i)} \tag{A.1}$$

where P and Q are the model and target probabilities, respectively, and i represents a microstate of the system. The probability of each microstate in the canonical ensemble can be expressed as

$$P(\boldsymbol{r}_i) = \frac{\mathrm{e}^{-\beta U(\boldsymbol{r}_i)}}{\int \mathrm{e}^{-\beta U(\boldsymbol{r})}\,\mathrm{d}\boldsymbol{r}^{\boldsymbol{N}}} \tag{A.2}$$

The overlap between two canonical ensembles (see ref 42 for more information) can be expressed as

$$\epsilon_{\mathrm{KL}} = \sum_i P_{\mathrm{AA}}(\mathbf{r}_i) \ln\left(\frac{P_{\mathrm{AA}}(\mathbf{r}_i)}{P_{\mathrm{CG}}(M(\mathbf{r}_i))}\right) + \langle S_{\mathrm{map}}\rangle_{\mathrm{AA}} \tag{A.3}$$

where $M(r)$ is the mapping operator from AA to CG configuration, S_{map} is the mapping entropy, which arises from degeneracies during the mapping of AA configuration, and $P_{\mathrm{AA}}(r_i)$ and $P_{\mathrm{CG}}(M(r_i))$ are the normalized probability of a configuration i in the AA and CG ensembles, respectively. Substituting eq A.2 into the above equation, the KL divergence can be rewritten as

$$\epsilon_{\mathrm{KL}} = \beta\langle U_{\mathrm{CG}} - U_{\mathrm{AA}}\rangle_{\mathrm{AA}} - \beta\langle F_{\mathrm{CG}} - F_{\mathrm{AA}}\rangle_{\mathrm{AA}} + \langle S_{\mathrm{map}}\rangle_{\mathrm{AA}} \tag{A.4}$$

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, respectively. The first derivative of the above equation with respect to the interaction parameters can be expressed as

$$\nabla\epsilon_{\mathrm{KL}}(\lambda_k) = \beta\left\langle \frac{\partial U}{\partial \lambda}\right\rangle_{\mathrm{CG}} - \beta\left\langle \frac{\partial U}{\partial \lambda}\right\rangle_{\mathrm{AA}} \tag{A.5}$$

which again can be simplified to

$$\nabla\epsilon_{\mathrm{KL}}(\lambda_k) = 2\pi\beta N\rho \int_0^\infty (g_{\mathrm{CG}}(r) - g_{\mathrm{AA}}(r))\frac{\partial u}{\partial \lambda} r^2 \, \mathrm{d}r \tag{A.6}$$

The optimality condition requires this quantity be zero. Furthermore, the curvature of KL distance (its Hessian) should be non-negative. The Hessian of KL divergence can be expressed as

$$\begin{aligned} H_{ij,\mathrm{KL}} = {} & \beta\left\langle \frac{\partial^2 U_{\mathrm{CG}}}{\partial \lambda_i \partial \lambda_j}\right\rangle_{\mathrm{AA}} - \beta\left\langle \frac{\partial^2 U_{\mathrm{CG}}}{\partial \lambda_i \partial \lambda_j}\right\rangle_{\mathrm{CG}} \\ & + \beta^2\left\langle \frac{\partial U_{\mathrm{CG}}}{\partial \lambda_i}\frac{\partial U_{\mathrm{CG}}}{\partial \lambda_j}\right\rangle_{\mathrm{CG}} - \beta^2\left\langle \frac{\partial U_{\mathrm{CG}}}{\partial \lambda_i}\right\rangle_{\mathrm{CG}}\left\langle \frac{\partial U_{\mathrm{CG}}}{\partial \lambda_j}\right\rangle_{\mathrm{CG}} \end{aligned} \tag{A.7}$$

where i and j are indices of second-order derivatives of KL divergence with respect to the parameters λ_i and λ_j, respectively. For linear parameters, which appear frequently in the empirical force fields, the Hessian of KL divergence reduces to

$$\mathbf{H} = \beta^2\left\langle \left(\frac{\partial U_{\mathrm{CG}}}{\partial \lambda}\right)^2\right\rangle_{\mathrm{CG}} - \beta^2\left\langle \frac{\partial U_{\mathrm{CG}}}{\partial \lambda}\right\rangle_{\mathrm{CG}}^2 \tag{A.8}$$

which denotes the variance of a derived quantity in the CG ensemble. Therefore, the curvature is always non-negative, implying the existence of a global minimum. Thus the problem considered here is a convex optimization problem in terms of the potential parameters, and, if a local minimum exists, then it has to be a global minimum.

To calculate the KL divergence, we employ the first-order forward finite difference algorithm over a grid representing discretization of the LJ potential parameters at the thermodynamic states of AAMD systems. The following formula is used for the finite difference method

$$\mathrm{d}\epsilon_{\mathrm{KL}} = \frac{\partial \epsilon_{\mathrm{KL}}}{\partial C_{12}}\mathrm{d}C_{12} + \frac{\partial \epsilon_{\mathrm{KL}}}{\partial C_6}\mathrm{d}C_6 \tag{A.9}$$

where the method is applied on a grid of size 150 × 150. The same mesh is used to compute the total variation in RDF.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge on the ACS Publications website at DOI: 10.1021/acs.jpclett.8b03872.

Additional material on the assessment and performance of the deep neural network as well as its implementation (PDF)

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: aluru@illinois.edu.

ORCID

Alireza Moradzadeh: 0000-0003-3355-7086

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

We thank Professor R. Srikant from the Electrical and Computer Engineering Department at the University of Illinois at Urbana−Champaign for helpful discussions. This work was supported by the National Science Foundation under grants 1506619, 1545907, 1708852, 1720633, and 1720701. We acknowledge the use of Blue Waters supercomputing resources at the University of Illinois at Urbana−Champaign. This work partially used the Extreme Science and Engineering Discovery Environment (XSEDE) Stampede2 at the Texas Advanced Computing Center through allocation TG-CDA100010.

■ REFERENCES

(1) 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* **2006**, *73* (6), 061201.

(2) Hospital, A.; Goñi, J. R.; Orozco, M.; Gelpí, J. L. Molecular Dynamics Simulations: Advances and Applications. *Adv. Appl. Bioinform. Chem.* **2015**, *8*, 37−47.

(3) 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.

(4) Henderson, R. L. A Uniqueness Theorem for Fluid Pair Correlation Functions. *Phys. Lett. A* **1974**, *49* (3), 197−198.

(5) 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: At., Mol., Opt. Phys.* **1986**, *33* (5), 3451−3465.

(6) Noid, W. G.; Chu, J.-W.; Ayton, G. S.; Voth, G. A. Multiscale Coarse-Graining and Structural Correlations: Connections to Liquid-State Theory. *J. Phys. Chem. B* **2007**, *111* (16), 4116−4127.

(7) D'Alessandro, M.; Cilloco, F. Information-Theory-Based Solution of the Inverse Problem in Classical Statistical Mechanics. *Phys. Rev. E* **2010**, *82* (2), 021128.

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

(9) Hansen-Goos, H.; Mecke, K. Fundamental Measure Theory for Inhomogeneous Fluids of Nonspherical Hard Particles. *Phys. Rev. Lett.* **2009**, *102* (1), 018302.

(10) Tang, Y.; Wu, J. Modeling Inhomogeneous van Der Waals Fluids Using an Analytical Direct Correlation Function. *Phys. Rev. E* **2004**, *70* (1), 011201.

(11) Rogers, F. J.; Young, D. A. New, Thermodynamically Consistent, Integral Equation for Simple Fluids. *Phys. Rev. A: At., Mol., Opt. Phys.* **1984**, *30* (2), 999−1007.

(12) Reith, D.; Pütz, M.; Müller-Plathe, F. Deriving Effective Mesoscale Potentials from Atomistic Simulations. *J. Comput. Chem.* **2003**, *24* (13), 1624−1636.

(13) 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.

(14) 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.

(15) 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.

(16) Byers, J. The Physics of Data. *Nat. Phys.* **2017**, *13* (8), 718−719.

(17) 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), 16576.

(18) 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), No. e1701816.

(19) 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.

(20) LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521* (7553), 436−444.

(21) Tabor, D. P.; Roch, L. M.; Saikin, S. K.; Kreisbeck, C.; Sheberla, D.; Montoya, J. H.; Dwaraknath, S.; Aykol, M.; Ortiz, C.; Tribukait, H.; et al. Accelerating the Discovery of Materials for Clean Energy in the Era of Smart Automation. *Nat. Rev. Mater.* **2018**, *3* (5), 5−20.

(22) 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.

(23) 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. Pharmaceutics* **2018**, *15* (10), 4386.

(24) 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.

(25) 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.

(26) 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.

(27) Carleo, G.; Troyer, M. Solving the Quantum Many-Body Problem with Artificial Neural Networks. *Science* **2017**, *355* (6325), 602−606.

(28) Nomura, Y.; Darmawan, A. S.; Yamaji, Y.; Imada, M. Restricted Boltzmann Machine Learning for Solving Strongly Correlated Quantum Systems. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2017**, *96* (20), 205152.

(29) van Nieuwenburg, E. P. L.; Liu, Y.-H.; Huber, S. D. Learning Phase Transitions by Confusion. *Nat. Phys.* **2017**, *13* (5), 435−439.

(30) Ch'ng, K.; Carrasquilla, J.; Melko, R. G.; Khatami, E. Machine Learning Phases of Strongly Correlated Fermions. *Phys. Rev. X* **2017**, *7* (3), 031038.

(31) Deng, D.-L.; Li, X.; Das Sarma, S. Quantum Entanglement in Neural Network States. *Phys. Rev. X* **2017**, *7* (2), 021021.

(32) Wang, H.; Yang, W. Force Field for Water Based on Neural Network. *J. Phys. Chem. Lett.* **2018**, *9* (12), 3232−3240.

(33) Bejagam, K. K.; Singh, S.; An, Y.; Deshmukh, S. A. Machine-Learned Coarse-Grained Models. *J. Phys. Chem. Lett.* **2018**, *9* (16), 4667−4672.

(34) Zhang, L.; Han, J.; Wang, H.; Car, R.; E, W. DeePCG: Constructing Coarse-Grained Models via Deep Neural Networks. *J. Chem. Phys.* **2018**, *149* (3), 034101.

(35) Abraham, M. J.; Murtola, T.; Schulz, R.; Páll, S.; Smith, J. C.; Hess, B.; Lindahl, E. ScienceDirect GROMACS : High Performance Molecular Simulations through Multi-Level Parallelism from Laptops to Supercomputers. *SoftwareX* **2015**, *2*, 19−25.

(36) 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.

(37) 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.

(38) Hornik, K.; Stinchcombe, M.; White, H. Multilayer Feedforward Networks Are Universal Approximators. *Neural Networks* **1989**, *2* (5), 359−366.

(39) Goh, G. B.; Hodas, N. O.; Vishnu, A. Deep Learning for Computational Chemistry. *J. Comput. Chem.* **2017**, *38* (16), 1291−1307.

(40) Kingma, D. P.; Lei Ba, J. ADAM: A Method for Stochastic Optimization. 2017. arXiv:1412.6980 [cs.LG]. arXiv.org e-Print archive. https://arxiv.org/abs/1412.6980.

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

(42) Chaimovich, A.; Shell, M. S. Coarse-Graining Errors and Numerical Optimization Using a Relative Entropy Framework. *J. Chem. Phys.* **2011**, *134* (9), 094112.

(43) Shell, M. S. The Relative Entropy Is Fundamental to Multiscale and Inverse Thermodynamic Problems. *J. Chem. Phys.* **2008**, *129* (14), 144108.

(44) Hyndman, R. J.; Koehler, A. B. Another Look at Measures of Forecast Accuracy. *Int. J. Forecast* **2006**, *22* (4), 679−688.