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 (C12, C6). The 12–6 LJ potential is prominently used in molecular dynamics (MD) simulations to study physical, chemical, biological, and mechanical systems. 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. As per Henderson’s theorem, 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. Different frameworks, such as the fundamental measure theory and integral equations, 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 or used to optimize statistical or empirical errors between the MD-calculated and given RDFs (e.g., relative entropy or the simplex minimization method). 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). 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, 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.
During the past decade, data-driven and machine-learning (ML) methods have received enormous attention in the computational analysis of physical problems.\textsuperscript{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\textsuperscript{20} in the modeling and understanding of physical phenomena such as data-driven materials discovery,\textsuperscript{21–26} calculation of the thermodynamic and ground states in multibody systems,\textsuperscript{27,28} phase transitions,\textsuperscript{29} classification of phases in strongly correlated Fermions,\textsuperscript{30} quantum entanglement,\textsuperscript{31}\textsuperscript{31} and many other applications.\textsuperscript{32}

During the last couple of years, researchers have started to develop deep neural networks (DNNs) for CG model development.\textsuperscript{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 $\mu$s 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 NN 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

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

(1)

<table>
<thead>
<tr>
<th>thermodynamic state</th>
<th>parameter range</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho$ (nm$^{-3}$)</td>
<td>$T$ (K)</td>
</tr>
<tr>
<td>$C_{12}$ (kJ mol$^{-1}$nm$^{-12}$)</td>
<td>$C_{6}$ (kJ mol$^{-1}$nm$^{-6}$)</td>
</tr>
<tr>
<td>min 8.0 290.0</td>
<td>0.000005 0.00005</td>
</tr>
<tr>
<td>max 19.4 400.0</td>
<td>0.0001 0.01</td>
</tr>
</tbody>
</table>

where $C_{12}$ ($= 4\epsilon\sigma^{12}$) and $C_{6}$ ($= 4\epsilon\sigma^{6}$) are the interaction potential parameters and $\epsilon$ and $\sigma$ 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 of system $i$ sampled uniformly over the range shown in Table 1. The LJ potential form is given by

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

(2)

where $f$ is a vector valued function, which is a nontrivial function of RDF, and thermodynamic state variables $T$ and $\rho$ 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 based on the universal approximation theorem, which states that FNN with enough capacity can approximate many continuous functions. The data of each MD simulation are fed into DNN as

$$x_i = (g(r), \rho_1^i, \rho_2^i, ..., \rho_p^i, T_i, T_1^i, ..., T_n^i)$$

(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 $i$th LJ system ($x_i \in D$). The first $n$ elements correspond to the value of RDF discretized between its minimum and maximum range, that is, $x_{i,j} = g(r_j) \forall j = 1, 2, ..., n$, and $r_j = r_{\text{min}} + (l - 1)dr$. The remaining 2$p$ elements of the input vector correspond to the different exponents of the scaled density and temperature; for example, $x_{i,n+j} = (\rho_1^i - \min \rho_1^i) / (\max \rho_1^i - \min \rho_1^i) \forall l = 1, 2, ..., p$, 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

$$f(x_i) = \phi_{x_i} \left( \sum_{j=1}^{n} W_j x_j + b_j \right) + b_0$$

(4)
where $\phi_k$ is the nonlinear activation function of layer $k$ given by

$$
\phi_k(x) = \phi_k(W_k \phi_{k-1}(x) + b_k) = \tanh(W_k \phi_{k-1}(x) + b_k) + \text{sigmoid}(W_k \phi_{k-1}(x) + b_k).
$$

Layer $k$ receives a linear transformation of the output of layer $k - 1$, given by $\phi_{k-1}(x)$, and applies the activation function $\phi_k$ to produce the output of layer $k$. Each layer, $k$, has weights, $W_k$, and bias, $b_k$, with dimensions, $W_k \in \mathbb{R}^{d_k \times d_{k-1}}$, and $b_k \in \mathbb{R}^{d_k}$, where $d_k$ is the number of nodes in layer $k$. $n_k$ denotes the number of hidden layers. The summation of the tangent hyperbolic and sigmoid nonlinearities ($\tanh x + \text{sigmoid} x$) is used between layers, except for the output layer ($\phi_n$) 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 predicted parameters is optimized. The adaptive moment estimation (Adam) optimizer is used to minimize the loss function ($L(\theta, D)$), which can be expressed as

$$
L(\theta, D) = \frac{1}{2|D|} \sum_{i \in D} \sum_{j \in \{C_i, C_j\}} (\nu_{ij}^\text{GT} - \nu_{ij}^\text{DNN}(\theta, x_i))^2
$$

where $\theta$ represents the free parameters (weights and biases) of DNN. $\nu_{ij}^\text{GT}$ and $\nu_{ij}^\text{DNN}(\theta)$ 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 data set. $|D|$ denotes the size of the training data set. The training data set corresponds to $\sim75\%$ of the data obtained from MD simulations, and it is used to find the optimal weights and biases of the DNN. The validation data set ($\sim12.5\%$ 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. The dropout technique drops nodes in layers with a specific probability, therefore training only a fraction of weights ($W_k$) and biases ($b_k$) 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 including 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 ($C_\phi$, $C_\rho$) 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.

The error in the total variation between CG and AA RDFs is calculated using the expression

$$
\epsilon_{\text{G} \to \text{AA}} = \frac{\int_0^\infty |g_{\text{CG}}(r) - g_{\text{AA}}(r)|^2 dr}{\int_0^\infty g_{\text{AA}}(r)^2 dr}
$$

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

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

where $U (= \sum_i u(r_{ij}))$ and $F$ are the potential energy and free energy of the system in the AA and CG systems and $\langle S_{\text{map}} \rangle_{\text{AA}}$ 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.

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
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,c 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. 99% of the training data are enclosed in this region. The region also encloses \(~99\%\) of the data of validation and testing data sets. The MAPE is also shown for each data set. (Red circles, square, pentagon, and diamond points represent training, validation, and 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.)

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 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.)
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).

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.

Table 2. Predictability of Argon Parameters on a $11 \times 11$ Mesh Covering Various Temperature and Density Values

<table>
<thead>
<tr>
<th>$C_{12} \times 10^{-6} \left[ \text{kJ mol}^{-1} \text{nm}^{-12} \right]$</th>
<th>ground truth</th>
<th>deep learning</th>
<th>relative error %</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.70</td>
<td>10.3 ± 0.116</td>
<td>6.18</td>
<td></td>
</tr>
<tr>
<td>$C_6 \times 10^{-3} \left[ \text{kJ mol}^{-1} \right]$</td>
<td>6.22</td>
<td>6.23 ± 0.0053</td>
<td>0.16</td>
</tr>
</tbody>
</table>

*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.
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, $\epsilon_{\text{rdf}}$ and $\epsilon_{\text{KL}}$. In Figure 5a,b, $\epsilon_{\text{rdf}}$ 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 $\epsilon_{\text{rdf}}$. The minimum point of $\epsilon_{\text{rdf}}$ lies in the vicinity of the LJ parameters predicted by the DNN. On the basis of eq 8, $\epsilon_{\text{KL}}$ between simple multiatom molecules and the LJ particles is shown in Figure 5e,f. Similar to $\epsilon_{\text{rdf}}$, the minimum value of $\epsilon_{\text{KL}}$ is close to the DNN-predicted parameters. The convexity of $\epsilon_{\text{KL}}$ ensures that the local minimum of $\epsilon_{\text{KL}}$ 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

$$
\epsilon_{\text{KL}}(P; Q) = -\sum_i P(i) \log \frac{Q(i)}{P(i)}
$$

(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

$$
P(r) = \frac{e^{-\beta U(r)}}{\int e^{-\beta U(r)} \, dr^N}
$$

(A.2)

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

**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 $\epsilon_{\text{rdf}}$ and $\epsilon_{\text{KL}}$ 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$_4$. 
The Journal of Physical Chemistry Letters

\[ \epsilon_{KL} = \sum_i P_{AA}(r_i) \ln \left( \frac{P_{AA}(r_i)}{P_{CG}(M(r_i))} \right) + \langle S_{map} \rangle_{AA} \]

(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 \(P_{AA}(r_i)\) and \(P_{CG}(M(r_i))\) 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

\[ \epsilon_{KL} = \beta(U_{CG} - U_{AA}) - \beta(F_{CG} - F_{AA}) + \langle S_{map} \rangle_{AA} \]

(A.4)

where \(U = \sum_i \mu(r_i)\) 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

\[ \nabla \epsilon_{KL}(\lambda_i) = \beta \left( \frac{\partial U}{\partial \lambda_i} \right)_{CG} - \beta \left( \frac{\partial U}{\partial \lambda_i} \right)_{AA} \]

(A.5)

which again can be simplified to

\[ \nabla \epsilon_{KL}(\lambda_i) = 2\pi \beta \rho_i \int_0^\infty (g_{CG}(r) - g_{AA}(r)) \frac{du}{dr} r^2 dr \]

(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

\[ H_{KL} = \beta \left( \frac{\partial^2 U_{CG}}{\partial \lambda_i \partial \lambda_j} \right)_{CG} - \beta \left( \frac{\partial^2 U_{CG}}{\partial \lambda_i \partial \lambda_j} \right)_{AA} + \beta^2 \left( \frac{\partial U_{CG}}{\partial \lambda_i} \right)_{CG} \left( \frac{\partial U_{CG}}{\partial \lambda_j} \right)_{CG} \]

(A.7)

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

\[ H = \beta^2 \left( \frac{\partial U_{CG}}{\partial \lambda_i} \right)^2_{CG} - \beta^2 \left( \frac{\partial U_{CG}}{\partial \lambda_i} \right)^2_{CG} \]

(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

\[ d\epsilon_{KL} = \frac{\partial \epsilon_{KL}}{\partial C_{12}} dC_{12} + \frac{\partial \epsilon_{KL}}{\partial C_{6}} dC_{6} \]

(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.jpcl.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

Applications and Prospects.


Problem with Artificial Neural Networks.

Representation of Molecules.


602

Network.


2018

Neural Computer for Molecular

Aladinskaya, A. V.; Aliper, A.; Zhavoronkov, A. Adversarial Threshold


(28) Nomura, Y.; Darmawan, A. S.; Yamaji, Y.; Imada, M. Restricted

Crossover Complexes with an Artificial Neural Network.

Era of Smart Automation.

et al. Accelerating the Discovery of Materials for Clean Energy in the

D.; Montoya, J. H.; Dwaraknath, S.; Aykol, M.; Ortiz, C.; Tribukait, H.;

(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.


(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-


(25) Kim, E.; Huang, K.; Jegelka, S.; Olivetti, E. Virtual Screening of

Inorganic Materials Synthesis Parameters with Deep Learning. npj


(26) Ramprasad, R.; Batra, R.; Pilania, G.; Mannodi-Kanakkithodi, A.;

Kim, C. Machine Learning in Materials Informatics: Recent


(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


96 (20), 205152.

(29) van Nieuwenburg, E. P. L.; Liu, Y.-H.; Huber, S. D. Learning


(30) Ch’ng, K.; Carraquilla, 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


(32) Wang, H.; Yang, W. Force Field for Water Based on Neural


(33) Bejagam, K. K.; Singh, S.; An, Y.; Deshmukh, S. A. Machine-


4667–4672.