The Journal of Chemical Physics

Machine learning approaches for structural and thermodynamic properties of a Lennard-Jones fluid

Cite as: J. Chem. Phys. **153**, 104502 (2020); https://doi.org/10.1063/5.0017894
Submitted: 11 June 2020 . Accepted: 17 August 2020 . Published Online: 08 September 2020

Galen T. Craven, Nicholas Lubbers, Kipton Barros, and Sergei Tretiak

AIP Publishing

The Journal of Chemical Physics ARTICLE scitation.org/journal/jcp

Machine learning approaches for structural and thermodynamic properties of a Lennard-Jones fluid

Cite as: J. Chem. Phys. 153, 104502 (2020); doi: 10.1063/5.0017894
Submitted: 11 June 2020 • Accepted: 17 August 2020 •
Published Online: 8 September 2020

Galen T. Craven,[1,a] Nicholas Lubbers,[2] Kipton Barros,[1] and Sergei Tretiak[3]

AFFILIATIONS

[1]Theoretical Division and Center for Nonlinear Studies (CNLS), Los Alamos National Laboratory, Los Alamos, New Mexico 87544, USA
[2]Computer, Computational, and Statistical Sciences Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87544, USA
[3]Theoretical Division, Center for Nonlinear Studies (CNLS), and Center for Integrated Nanotechnologies (CINT), Los Alamos National Laboratory, Los Alamos, New Mexico 87544, USA

Note: This paper is part of the JCP Special Topic on Machine Learning Meets Chemical Physics.
[a]**Author to whom correspondence should be addressed:** gcraven@lanl.gov

ABSTRACT

Predicting the functional properties of many molecular systems relies on understanding how atomistic interactions give rise to macroscale observables. However, current attempts to develop predictive models for the structural and thermodynamic properties of condensed-phase systems often rely on extensive parameter fitting to empirically selected functional forms whose effectiveness is limited to a narrow range of physical conditions. In this article, we illustrate how these traditional fitting paradigms can be superseded using machine learning. Specifically, we use the results of molecular dynamics simulations to train machine learning protocols that are able to produce the radial distribution function, pressure, and internal energy of a Lennard-Jones fluid with increased accuracy in comparison to previous theoretical methods. The radial distribution function is determined using a variant of the segmented linear regression with the multivariate function decomposition approach developed by Craven *et al.* [J. Phys. Chem. Lett. **11**, 4372 (2020)]. The pressure and internal energy are determined using expressions containing the learned radial distribution function and also a kernel ridge regression process that is trained directly on thermodynamic properties measured in simulation. The presented results suggest that the structural and thermodynamic properties of fluids may be determined more accurately through machine learning than through human-guided functional forms.

I. INTRODUCTION

The principal goal of statistical mechanics is to predict a system's collective behavior from its constituent interactions. In condensed-phase molecular systems, solving this problem is typically intractable analytically, and therefore, computer-assisted methods must be employed.[1,2] Over the last 60 years, molecular simulations, particularly with respect to fluids, have significantly advanced our understanding of how macroscale observables arise from interatomic interactions.[3–8] A major limitation of using computer simulations, however, is that they are typically performed in an *ad hoc* fashion and as such the predictive power of these models is limited. This is because the observations taken from computer simulations are virtual measurements of a system's properties under a particular set of physical conditions, and these measurements are, in general, not transferable to other physical conditions. In addition to the lack of transferability in the solutions produced by molecular simulations, these methods can also incur significant computational costs in order to make simple predictions. This is particularly true when the interatomic interactions are treated quantum mechanically, the system is very large, and/or the system must be simulated over a long timescale in order to make meaningful observations.[8–15]

Recently, new machine learning (ML) methods have been developed and applied to solve problems in physics and chemistry that were previously intractable using traditional molecular simulation methods.[16–20] The success of these ML methods has illustrated that many problems in the physical sciences can be solved faster and more accurately using data-driven approaches based on information analysis than through the application of protocols that institute some set of physics principles directly.[17,21–23] Machine learning methods are not without significant shortcomings, however. For example, if not trained and tested on a suitably general dataset, ML programs often fail catastrophically when asked to solve problems that require extrapolation outside of the dataset on which they are trained.[24–29] Despite having usability that is typically constrained to a narrow application window, ML approaches have been successfully applied to solve both new problems in physics and to develop improved solutions to old problems. The latter is the focus of this article in the context of a canonical problem in statistical mechanics—predicting the structural and thermodynamic properties of a Lennard-Jones (LJ) fluid.[30–37]

Previous attempts to model structural properties of the LJ fluid, specifically, the radial distribution function (RDF), have relied on fitting simulation data to empirically motivated functional forms[33–45] or using integral equation methods.[46–53] Classical density functional theories and quasi-continuum theories have also been developed to determine properties of simple fluids.[54] There are a number of empirical expressions in the literature that can be applied to produce the RDF of a LJ fluid.[33–37] The pioneering expression of Goldman[33] gives accurate predictions for the RDF in certain temperature and density regimes but has a limited range of applicability. The expression by Morsali *et al.*[35] gives accurate predictions for the RDF over a selected range of temperatures and densities, but it does suffer from the significant problem that its functional form is not continuous and therefore results in unphysical predictions. It also fails under low density conditions. Lack of continuity and a limited range of applicability are also problems in the expressions proposed by Bamdad *et al.* and Matteoli and Mansoori.[34,36] Other than fitting empirical functional forms, another common approach to predict the RDF of simple fluids is to use integral equation methods. However, integral equations have historically not produced acceptable results for the LJ fluid.[46–48] This is, in part, due to the fact that in order to produce accurate predictions, the functional form of the integral equation, i.e., the closure relation, must be modified specifically to treat the LJ system.[49–51,53]

Machine learning approaches have been applied to examine structural correlations in a LJ fluid. Moradzadeh and Aluru have trained a neural network to predict what values of the LJ parameters will give a specific form of the RDF. They then applied this deep learning approach to address inverse design problems for coarse-graining applications.[19] They have also trained an autoencoder network to determine the RDF of simple fluids from a limited number of atomistic configurations.[55] We have previously developed a ML method based on segmented linear regression and multivariate function decomposition that is able to predict the RDF of simple fluids with significantly increased accuracy in comparison to traditional theoretical approaches.[20]

Determining the equation of state (EOS) of a LJ fluid is another long-standing problem, which has attracted significant interest.[37,38,41–45,56] Although not an exhaustive list, prominent LJ equations have been developed by Nicolas *et al.*,[38] Johnson *et al.*,[41] Mecke *et al.*,[42] and Thol *et al.*,[44] among others. In each of these previous studies, the equation of state is constructed by fitting simulation data to an empirically motivated functional form. Comprehensive discussions of the advantages and disadvantages of many LJ equations of state can be found in Refs. 44 and 57. Recently, ML approaches have been applied to predict thermodynamic properties of fluid systems.[58] To our best knowledge, however, there have been no attempts to understand how ML can be applied to determine the thermodynamic properties of the LJ fluid and if these data-driven approaches provide any advantages compared to standard frameworks.

In this article, we illustrate how ML methods can be applied to determine properties of a LJ fluid with significantly increased accuracy in comparison to traditional human-guided fitting models and integral equation approaches. First, we provide a detailed analysis of the error decreases that can be expected in comparison to traditional theoretical methods when a modified version of the ML methodology developed in Ref. 20 is used to generate the RDF of a LJ fluid. Second, we use the learned RDF to determine the pressure and internal energy of a LJ fluid and compare those values to the values generated by several analytical functions. Finally, we train a kernel ridge regression (KRR) process[59–61] to determine the pressure and internal energy of a LJ fluid. Comparisons are made between the thermodynamic properties generated by each ML method and the values produced by two important EOS expressions for the LJ fluid. We find that the presented ML methods are able to produce properties of a LJ system more accurately in comparison to previous theoretical methods.

In general, the computational advantage of applying ML approaches to generate structural and thermodynamic properties of condensed-phase systems will be particularly significant when batches of data are needed for applications such as coarse-grained model development, analytical function fitting, inverse design approaches, and interfacing between scales in multi-scale physics applications. The most computationally expensive step in the implementation of the ML methods is the generation of training data, which is an upfront investment of computational effort. We have previously shown, however, that similar ML approaches to those applied here can produce significant error reductions in comparison to traditional theoretical methods even when trained on sparse amounts of data.[20] After training, the developed procedures produce structural and thermodynamic properties at a negligible computational cost in comparison to using molecular simulations.

The rest of this article is organized as follows: Sec. II A contains the details of the datasets that are used to train the different ML procedures. The technical details of the ML methods are presented in Sec. II B. A discussion of the testing datasets that are used to quantify the error generated by the developed methods is given in Sec. II C. The results of applying ML to predict the RDF, pressure, and internal energy of a LJ fluid are presented in Sec. III. Comparisons are made between the developed ML procedures, the results from integral equation methods, and several analytical models whose parameters are obtained by fitting simulation data to empirical functional forms. Concluding remarks and thoughts on the ramifications of this work are presented in Sec. IV.

II. METHODS

A. Training data

We use both home-generated and literature datasets to train our ML models. The data used to train the ML process to produce the LJ RDF was generated using MD simulations of a system of $N = 2500$ particles interacting through the cut and shifted LJ potential,

$$\phi_{cs}(r) = \begin{cases} \phi(r) - \phi(r_c), & r \le r_c \\ 0, & r > r_c, \end{cases} \tag{1}$$

with

$$\phi(r) = 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right], \tag{2}$$

where σ and ϵ are the standard LJ parameters and r_c is the cutoff distance. The values of the LJ parameters and the masses of the particles were chosen to correspond to argon. A cutoff of $r_c = 4.0\sigma$ was used, which is a typical cutoff chosen in LJ fluid studies, e.g., by Johnson *et al.* in Ref. 41. The simulations were performed in the NVT ensemble using the modified impulse Langevin integrator developed by Goga *et al.* in Ref. 62. A time step of $\Delta t = 1$ fs and a relaxation rate of $\gamma = 0.5$ ps^{-1} were used in the Langevin integrator. A neighbor list that was updated every ten time steps was employed to accelerate the sampling time. Standard periodic boundary conditions were used. The propagation of each MD trajectory was performed in three stages: In the initial equilibration stage lasting 5 ps, linear velocity scaling of each particle was used at every time step in order to maintain a constant temperature. In the second phase, which was the primary equilibration phase, the system was propagated for 500 ps using the modified impulse Langevin integrator with no sampling taking place. Finally, in the third phase, the system was propagated and sampled over 500 ps. During this phase, the RDF, pressure, and energy were measured.

Using the results of these simulations, we calculated the RDF $g(r^*, \rho^*, T^*)$ of the system as a function of the reduced LJ parameter's distance $r^* = r/\sigma$, density $\rho^* = \rho\sigma^3$, and temperature $T^* = k_B T/\epsilon$. The RDF was computed using a histogram bin width of 0.01 Å/$\sigma \approx 0.002\,94$. Simulations were performed at 629 state points on a grid in the $\rho^* \times T^*$ plane. The grid spacing was $\Delta\rho^* = 0.05$ and $\Delta T^* = 0.2$ in the respective dimensions. These points were mostly constrained to the vapor, liquid, and supercritical regions of the phase diagram, as shown in Fig. 1, although some of the sampled points were outside of the fluid region. The highest temperature and density sampled were $T^* = 6$ and $\rho^* = 1.25$, respectively.

We also trained KRR processes to determine the reduced pressure $P^* = P\sigma^3/\epsilon$ and reduced residual internal energy $U^* = U/N\epsilon$ of a LJ fluid. The data used to train these processes were generated by Gottschalk using Monte Carlo simulations.[56] The Gottschalk dataset contains thermodynamic properties of the LJ system from 8374 state points with temperatures $T^* \le 6.4$. Most of these state points are confined to the region shown in blue in Fig. 1. This dataset was used for training because it consisted of a large number of datapoints and it also contains a high density of points in the supercritical region of the phase diagram, a region that is often not well-sampled in other datasets. An extensive review and analysis of various other datasets containing thermodynamic properties of the LJ system can be found in Ref. 57.

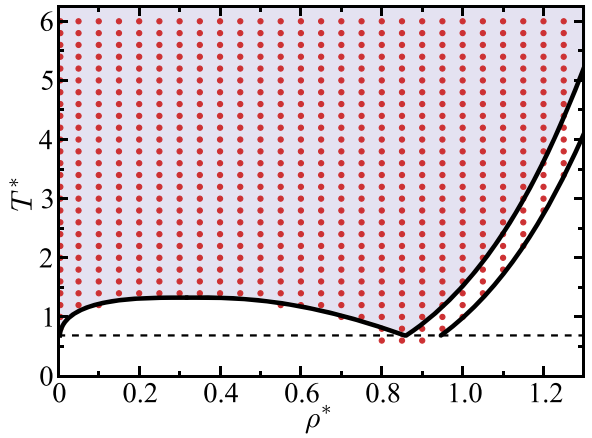

FIG. 1. State points in the $\rho^* \times T^*$ plane that were sampled to use as training data in the LR-RDF procedure. Each state point is shown as a red circle. The region encompassing the vapor, liquid, and supercritical fluid phases is shown in light blue and contains most of the red dots. The vapor–liquid and fluid–solid coexistence curves are shown as solid black lines. The triple point temperature ($T^*_{tp} = 0.687$) is shown as a dashed black line. All of the data used to construct the coexistence curves were taken from Ref. 45 and references therein.

B. Machine learning procedures

1. Radial distribution function

The ML method we applied to determine the LJ RDF is a variant of the linear regression (LR) with the multivariate function decomposition approach developed in Ref. 20. Here, we term this approach the LR-RDF method. The MD data used to train this process consisted of feature vectors of the form $\{r_k^*, \rho_i^*, T_i^*\}$, which were mapped to labels $g(r_k^*, \rho_i^*, T_i^*)$, where r_k^* is a particular value of the variable r^* and $p_i = \{\rho_i^*, T_i^*\}$ is a state point in the training data.

The implementation algorithm for this method is as follows: First, take an input state point $p = \{\rho^*, T^*\}$ (the point where the RDF would like to be predicted) and determine the $N_{neigh} = 4$ nearest neighbor points in the training data to the input point. The metric used to determine the distance between input point p and training point p_i is the weighted Euclidean distance

$$d_i = \sqrt{w_1(\rho_i^* - \rho^*)^2 + w_2(T_i^* - T^*)^2}, \tag{3}$$

with weights $w_1 = 1$ and $w_2 = 0.035$, which were obtained using a grid search hyperparameter optimization. Next, take the discretized training data in the r^* variable, and at each particular value, $r^* = r_k^*$ construct a linear approximation to the RDF,

$$\begin{aligned} g(r_k^*, \rho^*, T^*) &\approx \hat{g}_k(r_k^*, \rho^*, T^*) \\ &= a_0(r_k^*) + a_1(r_k^*)\rho^* + a_2(r_k^*)T^*, \end{aligned} \tag{4}$$

where $\hat{g}_k$ is a linear function in ρ^* and T^* with coefficients that are determined from multivariate least squares regression. The linear

approximation is based on the assumption that in the local region in $\rho^* \times T^*$ space around input point p, the RDF will vary approximately linearly in ρ^* and T^* at each r_k^*. Finally, combine the collection of regression processes at each r_k^* to construct the total RDF at point p,

$$g(r^*,\rho^*,T^*) \approx \mathcal{I}\left[\hat{g}_1(r_1^*,\rho^*,T^*),\hat{g}_2(r_2^*,\rho^*,T^*),\ldots,\hat{g}_K(r_K^*,\rho^*,T^*)\right], \tag{5}$$

where $\mathcal{I}$ is an interpolant over the variable r^* between functions in the set $\mathbf{G} = \{\hat{g}_1, \hat{g}_2, \ldots, \hat{g}_K\}$ and K is the index of the maximum r^* value used in the regression. Structural correlations in the LJ fluid occur over length scales in the dimensional r variable commensurate with σ. Therefore, if the spacing of the training data in the $r^* = r/\sigma$ dimension is $\ll 1$, the choice of the functional form of the interpolant $\mathcal{I}$ (linear, polynomial, etc.) will not be significant. Moreover, for most numerical applications, knowledge of the discretized set of functions $\mathbf{G}$ will be sufficient, and therefore, interpolation over r^* will not be necessary. Because the core regression process used in the LR-RDF method is two-dimensional linear regression that is fit to four neighboring state points, there is no significant overfitting in the procedure.

2. Equation of state

We developed an equation of state for the LJ fluid by training KRR processes to predict the thermodynamic properties pressure P^* and internal energy U^*. The training data for these processes consisted of feature vectors (state points) $p_i = \{\rho_i^*, T_i^*\}$, which were mapped to corresponding labels $P^*(\rho_i^*, T_i^*)$ and $U^*(\rho_i^*, T_i^*)$. The data used for training was taken from the Gottschalk dataset.[56] This dataset is well-suited to use for training because of its size and because the cutoff distance in the simulations used to generate the data is large. A proximity-based approach was employed in the KRR training procedures, meaning that for a particular input point p, only local training data was used to determine the pressure and internal energy at that point. Specifically, the training data used to generate P^* and U^* at point p consisted of $N_{\text{neigh}} = 16$ nearest neighbor points in the training data as quantified through the distance metric in Eq. (3). The number of neighboring points used in the regression process was found through hyperparameter optimization.

The KRR procedure was implemented using the scikit-learn ML package for Python.[63] A polynomial kernel

$$K(p_i,p_j) = \left(\gamma p_i^{\top} p_j + c_0\right)^d, \tag{6}$$

of degree $d = 4$ was employed to quantify the similarity between state points in the training data. For each regression process, we performed optimization of the c_0 parameter in Eq. (6) using a grid search where the optimization scoring function was the absolute percent error of the corresponding thermodynamic property P^* or U^*. The conditioning factor α in the KRR process (see Ref. 63 and associated documentation) was set to unity. In each regression, we used a value of $\gamma = 100$ in Eq. (6), which was found through a grid search. We took γ to be constant because we found that its optimized value would typically not deviate significantly from the given constant value and because doing so decreases the computational time needed to perform the KRR. In comparison, we found that the optimal value of c_0 would change significantly for each regression process, and therefore, a grid search optimization of this parameter was performed during each KRR procedure.

C. Test data

The test data for the ML procedures were obtained using the same simulation protocol described in Sec. II A. We generated two different test sets. The first test set was generated by performing MD simulations at 300 random points uniformly distributed over the region $[0.05, 1.15] \times [1.0, 5.0]$ in the $\rho^* \times T^*$ plane and measuring the RDF, pressure, and internal energy at each point. In order to account for errors incurred by using the cut and shifted potential in Eq. (1), standard tail corrections were applied to the measured pressure and internal energy values.[1,41] Any points outside of the vapor, fluid, or supercritical regions were removed from the test set, leaving 265 points. The second test set was generated by performing simulations along select isodensity contours ($\rho^* = \text{const.}$) using a spacing of $\Delta T^* = 0.2$ in the T^* dimension. For each contour, the lowest temperature sampled was dictated by the fluid phase boundaries shown in Fig. 1, and the highest temperature sampled was $T^* = 5$.

Additionally, there are multiple datasets in the literature containing thermodynamic properties of the LJ system at various state points measured using molecular simulation methods. Here, we used the datasets generated by Johnson *et al.*, Stephan *et al.*, and Meier as test sets to further gauge the accuracy of the developed ML methods.[41,57,64] In each of these datasets, we removed any outlier points as identified in the work of Ref. 57, any points that were outside of the fluid region of the phase diagram (the blue region in Fig. 1), and any points with temperature $T^* > 6$. After this procedure, the trimmed Johnson, Stephan, and Meier datasets consisted of 134, 317, and 269 points, respectively. We chose to use these datasets as test sets because they are some of the largest datasets available for the LJ fluid and also the cutoff distance used in each set is large ($r_c \geq 4.0\sigma$).

III. RESULTS AND DISCUSSION

Most previous work developing predictive models for the RDF of the LJ fluid has focused on fitting the results of simulations to empirically selected analytical forms or applying integral equation approaches. The main advantage of using empirical models is that they allow fast and often accurate approximation of the RDF of a LJ fluid without the need to perform computationally taxing simulations or numerical procedures. The ML methodology we apply here can produce LJ RDFs at a negligible computational cost, but with improved accuracy in comparison to human-guided procedures and integral equations.

Comparisons between RDFs measured in simulation and RDFs predicted by the ML procedure described in Sec. II B are shown in Fig. 2 for various temperature and density values. Excellent agreement is observed between the ML and MD results. The results for high-density systems with strong structural correlations are shown in Figs. 2(a) and 2(b). In these systems, the ML protocol almost exactly captures the highly structured form of the RDF. As shown

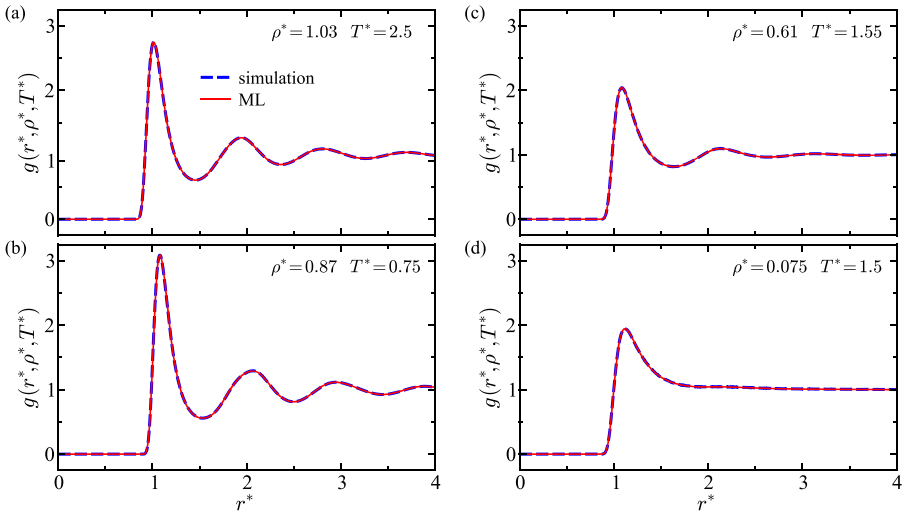

FIG. 2. Radial distribution functions predicted by ML (solid red) and measured in simulation using MD (dashed blue). Note that the two curves are almost indistinguishable and their superposition may appear pink. Results are shown for various state points: (a) $\rho^* = 1.03$; $T^* = 2.5$, (b) $\rho^* = 0.87$; $T^* = 0.75$, (c) $\rho^* = 0.61$; $T^* = 1.55$, and (d) $\rho^* = 0.075$; $T^* = 1.5$.

in Figs. 2(c) and 2(d), the ML method also generates accurate predictions for the RDF in intermediate- and low-density systems. This is notable because, as discussed in detail below, previous analytical expressions for the LJ RDF typically fail to give accurate predictions at low densities. In all cases, the ML results are in excellent agreement with the simulation results, illustrating the power of using ML to generate structural properties of fluids.

Figure 3 illustrates the root mean square deviation (RMSD) of the RDF,

$$\text{RMSD} \equiv \sqrt{\frac{\sum_{k=1}^{K_c}\left(g_{\text{MD}}(r_k^*,\rho^*,T^*)-g_{\text{theory}}(r_k^*,\rho^*,T^*)\right)^2}{K_c}}, \quad (7)$$

predicted using various theoretical methods g_{theory} taken with respect to the RDF measured in simulation g_{MD} along several isodensity contours. The RMSD was computed at $K_c = 1362$ points along the variable r^* using an equidistant spacing of ~0.002 94 between points. This corresponded to a maximum value of $r^* \approx 4$ used in the RMSD calculation. We truncated the RMSD calculation at this value to avoid sampling the structureless regime $r^* \to \infty$ where $g(r^*, \rho^*, T^*) = 1$ as sampling this regime would bias the RMSD toward a lower value. The lower bound for T^* along each contour is dictated by the fluid phase boundaries shown in Fig. 1.

Along the isodensity contour $\rho^* = 1.11$, shown in Fig. 3(a), the RMSD of the Morsali expression is ~0.025 while the RMSD of the Goldman expression is ~0.02. In contrast, the RMSD generated using ML varies from 0.0005 to 0.001. We also tested if approximating the RDF using the data from the nearest neighbor point in the training set yields an accurate estimate for the RDF. The nearest neighbor approximation yields a RMSD of ≈0.01. This illustrates that the accuracy of the ML method arises from the fitting

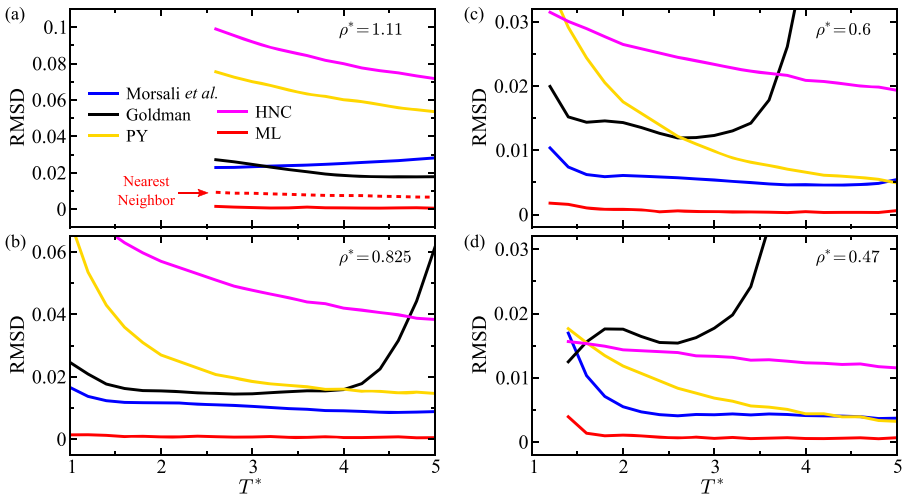

FIG. 3. Root mean square deviation of the RDF generated by ML (red), the expressions of Goldman (black) and Morsali *et al.* (blue), and the solutions to the PY (yellow) and HNC (magenta) equations, all shown as a function T^* along the isodensity contours: (a) $\rho^* = 1.11$, (b) $\rho^* = 0.825$, (c) $\rho^* = 0.6$, and (d) $\rho^* = 0.47$. The red dashed curve in panel (a) is the result given by approximating the RDF at the input point using the RDF of the nearest neighbor point on the training grid.

procedure and is not simply a function of the density of training data. The ML approach also gives more accurate results with respect to well-known integral equations. Specifically, the ML RMSD is greater than an order of magnitude lower than solutions to the Percus–Yevick (PY) and hypernetted-chain equations (HNC). The solutions to the integral equations were obtained using the pyPRISM program.[65]

The results along the contour $\rho^* = 0.825$ are shown in Fig. 3(b). The Morsali expression results in a RMSD of ~0.01, while the Goldman expression produces an RMSD that is ~0.015 for low T^* but diverges as T^* is increased. The RMSDs produced by the PY and HNC solutions are large for low T^* but decrease to ~0.015 and 0.04, respectively, as the temperature is increased. The ML result ranges between ~0.0005 to 0.001, greater than an order of magnitude decrease when compared to the RMSDs obtained using other methods.

Figure 3(c) illustrates the results for $\rho^* = 0.6$, which was a density included in the training set. We have used this density to illustrate the power of ML in comparison to other methods at a commonly examined density, not to infer any accuracy metrics about the ML procedure itself. The RMSD due to the Morsali expression is ~0.005. The Goldman expression produces an RMSD of ~0.015 for $T^* < 4$ but diverges for $T^* > 4$. The PY and HNC solutions generate errors that are, in general, greater than the other methods. In comparison, the RMSD due to ML is ~0.001—a significant error reduction.

The results along the contour $\rho^* = 0.47$ are shown in Fig. 3(d). The Morsali RMSD is ~0.02 at low temperatures and decreases to ~0.004 at high temperatures. The Goldman expression again produces an RMSD result of ~0.015 for low values of T^* but quickly diverges as T^* is increased. Away from the low-temperature regime, the integral equations result in errors of the order 0.005 and 0.012 for the PY and HNC solutions, respectively. The ML RMSD is ~0.004 at low temperatures but then decreases quickly to ~0.0007 as T^* is increased. Typically, we found that using ML to predict the LJ RDF resulted in greater than an order of magnitude decrease in the RMSD in comparison to traditional theoretical methods in high- and intermediate-density regimes.

The RMSD calculated along the isodensity contours $\rho^* = 0.25$ and $\rho^* = 0.125$, which both correspond to the low-density regime of the LJ fluid, are shown in Figs. 4(a) and 4(b) respectively. Notice that the RMSD for each density is shown on a log scale. In both cases, the ML protocol developed in this article significantly outperforms both the Morsali and Goldman expressions. In Fig. 4(a), the Morsali RMSD is of the order 1 for low T^* but quickly decreases to 0.1 as T^* is increased, while the Goldman RMSD is of the order 0.1 at lower temperatures but diverges to values greater than 100 for $T^* > 4$. In comparison, the RMSD calculated using ML is ~0.002 at low T^* but decreases to ~0.0005 as the temperature increases. For the case of $\rho^* = 0.125$, shown in Fig. 4(b), the Morsali expression fails catastrophically and produces an RMSD that is, at best, of the order 10, while the Goldman RMSD is again ~0.1 for $T^* < 4$ but diverges for $T^* > 4$. The failure of the Morsali expression arises because it is a piecewise noncontinuous function, and one of the branches of the expression diverges at low density creating the large deviations from the MD results, using the ML results in an RMSD of ~0.002 at this density.

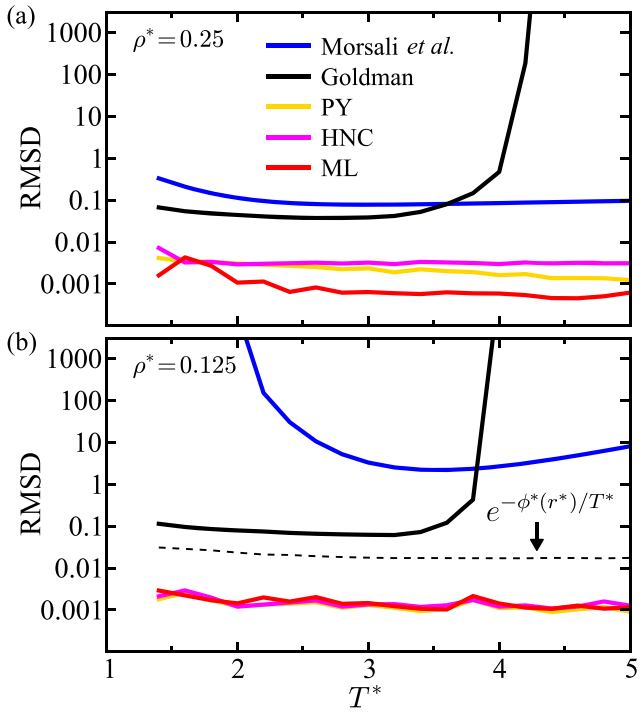

FIG. 4. Root mean square deviation of the RDF generated by ML (red), the expressions of Goldman (black) and Morsali *et al.* (blue), and solutions to the PY (yellow) and HNC (magenta) equations. The RMSD values are shown on a log scale. Each result is shown a function T^* along the isodensity contours (a) $\rho^* = 0.25$ and (b) $\rho^* = 0.125$. The dashed black curve in panel (b) is the result calculated using the theoretical RDF $g(r^*, T^*) = e^{-\phi^*(r^*)/T^*}$ corresponding to the infinite dilution limit.

In contrast to the empirically motivated functional forms, both the PY and HNC integral equations perform reasonably well at low density. For $\rho^* = 0.25$, shown in Fig. 4(a), the PY and HNC equations produce RMSDs that are ~2 to 5 times greater than ML. Therefore, the integral equation RMSDs are large in comparison to the ML method but provide a dramatic improvement over both the Goldman and Morsali expressions. For $\rho^* = 0.125$, shown in Fig. 4(b), the RMSD produced by both integral equations is approximately equal to that produced using ML. Also shown in Fig. 4(b) is the result given by taking the theoretical RDF to correspond to the infinite dilution limit $g(r^*, T^*) = e^{-\phi^*(r^*)/T^*}$, with $\phi^*(r^*)/T^* = \phi(r)/k_B T = 4/T^*[(1/r^*)^{12} - (1/r^*)^6)]$. Infinite dilution corresponds to the limit in which multi-body effects vanish. The RMSD given by infinite dilution approximation is approximately an order of magnitude greater than the RMSD generated by ML. This illustrates that theoretical procedures such as ML or integral equation methods must be employed to obtain accurate estimates for the RDF in systems with significant multi-body structural correlations. In general, we find that using ML to predict the RDF in low-density LJ systems will typically provide two orders of magnitude or greater decrease in the RMSD when compared to empirical functional forms and will often improve on the results of integral equation solutions.

Shown in Table I is the RMSD generated by each theoretical method calculated over the random test set described in Sec. II C. The first column lists the various theoretical methods. The second column is the RMSD result for each method calculated over every point in the test set. The ML approach reduces the error by factors of ~20 and 30 in comparison to the PY and HNC equations, respectively. The results are not shown for the Morsali and Goldman expressions because the claimed range of validity of both expressions is $\rho^* > 0.35$, and the lower density bound in the random test set is $\rho^* = 0.05$. We therefore cannot assess the error generated by these expressions using the full test set because both expressions return very large and often numerically divergent RMSD values in the low density regime (see Fig. 4). In order to construct a test set to compare the theoretical methods away from the low density regime, we removed any points from the full set with $\rho^* < 0.35$ and calculated the RMSD generated by each method over this trimmed dataset. These results are shown in the third column of the table. Over the trimmed test set, ML again generates the lowest error. The error from the Morsali expression is the next lowest, roughly 15 times greater than ML. The ML approach reduces the error by respective factors of ~40 and 60 in comparison to the PY and HNC equations. Note that the RMSD generated by each integral equation increases for the trimmed test set in comparison to the full test set because integral equations perform best in the low-density regime, and therefore, removing the low-density state points increases the average RMSD. The Goldman expression returns divergent RMSD values in the high-temperature regime, as shown in Fig. 3. Therefore, to calculate the average RMSD for the Goldman expression, we also removed any state points from the test set that generated an RMSD > 1. The RMSD value generated by ML over the trimmed test set is approximately a factor 60 lower than RMSD calculated using the Goldman expression with high-temperature state points removed. The RDF can also be estimated using data from the nearest point in the training data set. Over both the full and trimmed test sets, the ML method reduces the RMSD by approximately an order of magnitude in comparison to using the RDF from the nearest neighbor to estimate the RDF at the input point.

Shown in Fig. 5 is the learning curve for the LR-RDF method illustrating how the accuracy of the method varies as a function of the number of points in the training set. When using as few as 100–200 training points, ML reduces the RMSD by approximately an order of magnitude, or more, in comparison to each of the other methods. Moreover, the LR-RDF method outperforms all the other methods when trained on as few as ~30 points. Approximating the RDF at the input point using data from the nearest neighbor point in the training set yields better results than all the other methods except ML when the number of training points, i.e., the density of training data, is large but yields poor results in comparison with the other methods as the training data becomes sparse. It should be noted that as the density of the training data decreases, a threshold is reached where the ML method ceases to be more accurate than other theoretical methods. This threshold is reached, however, when there are only a few points in the training set.

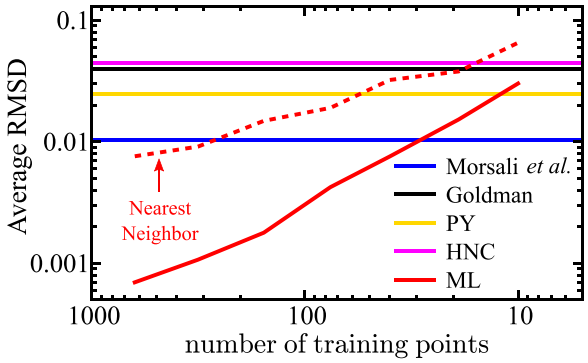

FIG. 5. Learning curve for the LR-RDF ML method illustrating the average RMSD of the trimmed dataset from the third column in Table I as a function of the number of training points. Results are shown for the LR-RDF procedure (red), the expressions of Goldman (black) and Morsali *et al.* (blue), and solutions to the PY (yellow) and HNC (magenta) equations. The red dashed curve is the result given by approximating the RDF at the input point using the RDF of the nearest neighbor point on the training grid. Both axes are shown on a log scale.

TABLE I. Average RMSD of the RDFs generated by various methods calculated over the random test set. The first column of RMSD values includes all the test points. The second column includes only test points with $\rho^* > 0.35$, which is the reported range of validity of the Morsali and Goldman expressions.

Method	Average RMSD	Average RMSD (excluding low density)[a]
ML	0.0009	0.0007
Morsali *et al.*	...	0.0104
Goldman	...	0.0394[b]
PY	0.0171	0.0246
HNC	0.0307	0.0441
Nearest neighbor	0.0066	0.0076

[a]Test points with $\rho^* < 0.35$ are excluded.
[b]Test points that generated an RMSD > 1 are excluded.

It is important to note that the different analytical expressions developed to generate the LJ RDF have been fit to data that were obtained using many different simulation protocols.[33,35,36,38] Because of these discrepancies between simulation procedures, we can only make observations about how well any previously developed functions are able to reproduce the data obtained under the specific simulation conditions used here. However, when the results of these prior analytical expressions are compared to our simulation data, we observe RMSD values that are typically similar to or less than those reported in the original articles. This implies that the simulation data obtained in this manuscript is well-representative of the data used to fit these functions. Moreover, based on the observed accuracy of the developed ML methods, we expect that ML will provide substantial improvements when compared to all of the previous empirical functions over a broad range of simulation conditions.

The major bottleneck of applying the present ML method is generating training data in a way that effectively samples the fluid region of the phase diagram while also being computationally efficient. While the comparisons given here illustrate the improvements

in predictive accuracy that can be expected when using ML to generate the RDF of a simple LJ fluid (see also the supplementary material of Ref. 20), questions remain with respect to the best way to generate training data and if a grid-based approach will scale well when applying the developed ML method to determine structural properties of more complex systems such as fluid mixtures. These questions will be addressed in our future work.

The ability to accurately predict macroscopic thermodynamic quantities from the function used to generate the RDF is an important test of both the precision of the fitting procedure and the validity of the chosen functional form used in the fitting.[33–37] In simple fluids that are dominated by pairwise interactions, structurally based macroscopic observables can be expressed using standard thermodynamic relations containing the RDF. In the specific case of a LJ fluid, the reduced pressure $P^*(\rho^*, T^*)$ and reduced internal energy $U^*(\rho^*, T^*)$ are related to the RDF through the respective expressions[1]

$$P^*(\rho^*, T^*) = \rho^* T^* - 16\pi\rho^{*2} \int_0^\infty (r^{*-4} - 2r^{*-10}) g(r^*, \rho^*, T^*)\, dr^* \tag{8}$$

and

$$U^*(\rho^*, T^*) = 8\pi\rho^* \int_0^\infty (r^{*-10} - r^{*-4}) g(r^*, \rho^*, T^*)\, dr^*. \tag{9}$$

Therefore, once a functional form for the RDF is known, it provides a direct route to generate the thermodynamic properties of the LJ system.

To further gauge the accuracy of the developed ML protocol, we evaluated Eqs. (8) and (9) numerically using the learned RDF and compared those results to pressure and internal energy values that were measured directly in MD simulations. Calculating macroscopic observables using the predicted RDF and comparing those values to values measured in simulation is a standard method that is typically used to test the accuracy of the specific procedure that is applied to generate the LJ RDF. We applied a long-range correction to the ML RDF by taking $g(r^*, \rho^*, T^*) = 1$ for values of $r^* > L/2$, where L is the simulation box length in the MD training data.

Figure 6 illustrates a comparison between the results for P^* and U^* measured directly in simulation and the results calculated by using the ML RDFs in Eqs. (8) and (9). The pressure and internal energy results are shown in Figs. 6(a) and 6(b), respectively. In both figures, the results are shown as a function of T^* along various isodensity contours. The agreement between the two methods is excellent in all cases.

Shown in Fig. 7 is the percent error,

$$X_{\text{error}}(\rho^*, T^*) = \frac{X_{\text{theory}}(\rho^*, T^*) - X_{\text{MD}}(\rho^*, T^*)}{X_{\text{MD}}(\rho^*, T^*)} \times 100, \tag{10}$$

for each thermodynamic property $X \in \{P^*, U^*\}$ calculated along various isodensity contours using the LR-RDF method as well as the Goldman and Morsali expressions. In Eq. (10), X_{theory} is the value predicted by the particular theoretical method, and X_{MD} is the value taken from MD simulation. Figures 7(a) and 7(b) show P^*_{error} along the contours $\rho^* = 1.021$ and $\rho^* = 0.88$, respectively. The ML procedure generates significantly lower error than the other methods at these high densities. It is interesting to note that the error generated by the Goldman and Morsali expressions is quasi-oscillatory, while the error due to the ML approach presents as noise. As shown in Fig. 7(c), the error reduction produced by using the LR-RDF method is particularly pronounced in the low-density regime. Specifically, for $\rho^* = 0.25$, the error produced by the Goldman expression is very large, while the Morsali expression produces an error of ≈10%. In contrast, ML produces an error of ≈0.06%. The percent error in the internal energy U^*_{error} along the contours $\rho^* = 1.021$ and $\rho^* = 0.88$ are, respectively, shown in Figs. 7(d) and 7(e). The LR-RDF method outperforms the other expressions, particularly at high temperatures. Figure 7(f) shows the internal energy error for $\rho^* = 0.25$ where, again, applying ML produces significantly lower error than the other methods.

In general, the absolute error $|X_{\text{error}}(\rho^*, T^*)|$ in each thermodynamic property calculated using the LR-RDF method is typically in the range 0.05%–0.5% across all density regimes. In contrast, both the Morsali and Goldman expressions result in average absolute errors of ~2% at high densities and greater than 10% at low densities. These results highlight the effectiveness of applying ML to predict a system's macroscopic properties using the learned RDF.

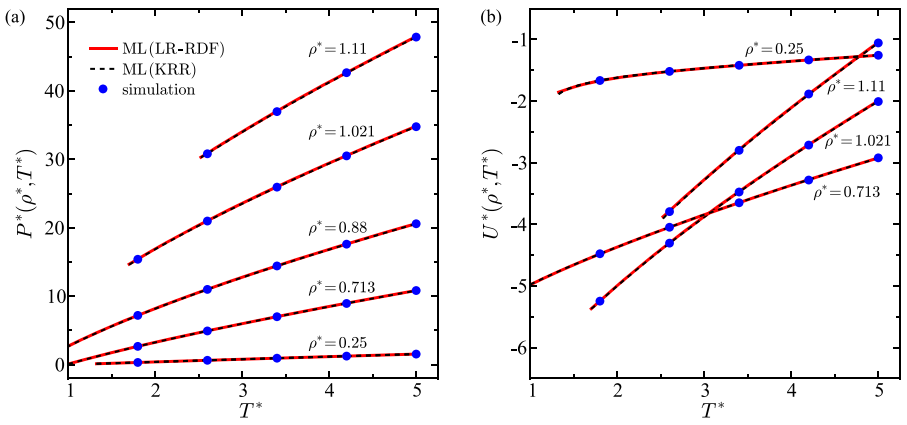

FIG. 6. Pressure and internal energy values calculated along various isodensity contours are shown in panels (a) and (b), respectively. Results are shown for densities $\rho^* = 1.11$, $\rho^* = 1.021$, $\rho^* = 0.88$ [excluded in panel (b) for visual clarity], $\rho^* = 0.713$, and $\rho^* = 0.25$. The blue circles are the results measured directly in MD simulations. The red curves are the results predicted from ML using the LR-RDF method, and the dashed black curves are the ML results obtained using the KRR approach.

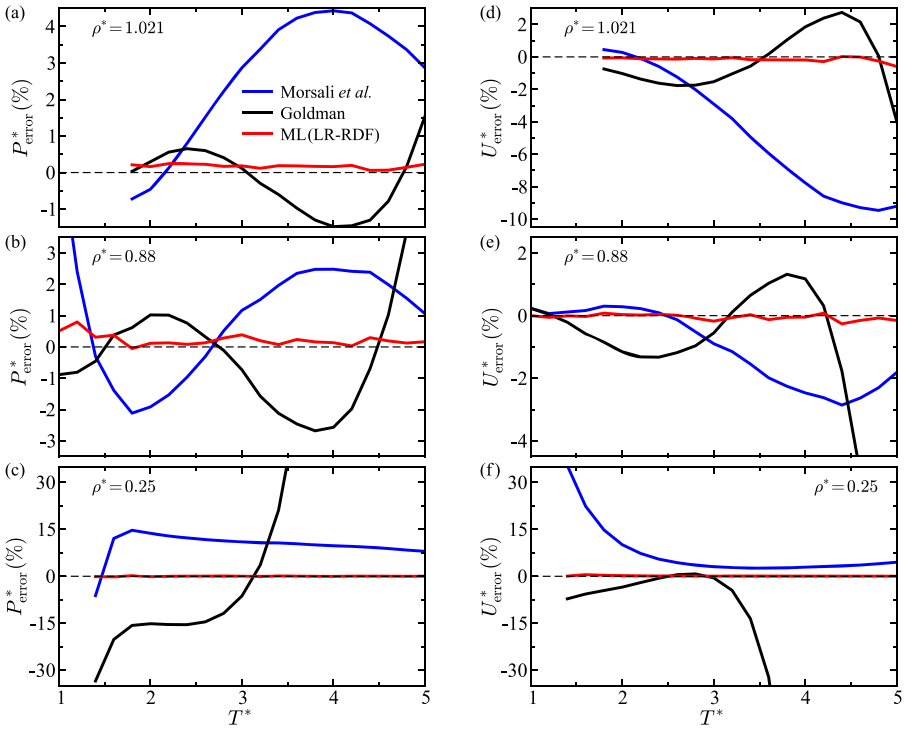

FIG. 7. Percent error in the pressure (left) and internal energy (right) calculated using the RDF generated by the ML(LR-RDF) method (red) and the expressions of Goldman (black) and Morsali *et al.* (blue). The results are shown as a function of temperature along the isodensity contours $\rho^* = 1.021$ [(a) and (d)], $\rho^* = 0.88$ [(b) and (e)], and $\rho^* = 0.25$ [(c) and (f)]. The dashed line in each panel denotes zero error and is shown for visual comparison.

We also trained a KRR process to produce thermodynamic properties of the LJ system as described in Sec. II B 2. This process is trained directly on thermodynamic properties measured in simulation and therefore does not include information from the RDF. The black dashed lines in Fig. 6 are the results for P^* and U^* generated using KRR along various isodensity contours. In all cases, the agreement between the MD results and the results generated using ML is excellent. In fact, upon detailed examination, it can be observed that the KRR process yields results that are in better agreement with the MD values than those generated using the LR-RDF method.

Figure 8 illustrates a comparison between the percent error in the pressure and internal energy values generated using the KRR method, the LR-RDF method, the Johnson EOS, and the Mecke EOS. The results are shown along various isodensity contours. The errors in the predicted pressure for high-density systems with $\rho^* = 1.021$ and $\rho^* = 0.88$ are shown in Figs. 8(a) and 8(b), respectively. The KRR method is the most accurate in both cases, generating an error along each contour that fluctuates around zero. The errors generated by all of the theoretical methods are correlated because the same MD data are used to compute the error for each method. The MD data fluctuates along each isodensity contour due to sampling errors, and so calculating the percent error generated by each theoretical equation of state produces correlated fluctuations. The results for a low-density system with $\rho^* = 0.25$ are shown in Fig. 8(c). In this case, both the KRR and LR-RDF methods generate errors along the contour that fluctuate about zero, while the Mecke and Johnson expressions yield respective errors of ~0.25% and 0.5%. The percent error in the internal energy produced by each method for $\rho^* = 1.021$ and $\rho^* = 0.88$ is shown in Figs. 8(d) and 8(e). At these high densities, the Johnson EOS is clearly not as accurate as the other methods and predicts values for the internal energy that varies significantly from the MD results. The Mecke EOS and the LR-RDF method yield similar errors at both densities, while the KRR method is again the most accurate. The error in the predicted internal energy for $\rho^* = 0.25$ is shown in Fig. 8(f). The Mecke and Johnson expressions produce errors that change sign as T^* is increased. Therefore, while the average percent error along the contour is approximately zero, the *absolute* percent error generated by each expression will be relatively large. The LR-RDF method yields slightly better results than the two analytical EOSs, and the KRR process is, again, the most accurate method. The KRR method and the Mecke and Johnson expressions are fit to two-dimensional input data while the LR-RDF method uses three-dimensional inputs. It is therefore noteworthy that the LR-RDF procedure produces comparable results to the others while being trained on more complex data structures.

Table II shows the results of using various methods to calculate the average absolute percent error in the pressure $\langle |P^*_{\text{error}}| \rangle$ over four different test sets. The specific methods compared are the KRR and LR-RDF ML methods, the Johnson and Mecke EOS expressions, and the result obtained by approximating the pressure at the input point using the pressure value from the nearest neighbor point in the training set. The examined test sets are the random test set generated here and the datasets of Johnson *et al.*, Stephan *et al.*, and Meier.[41,57,64] Descriptions of each test set can be found in Sec. II C. The combined error averaged over all of the test sets is also shown in

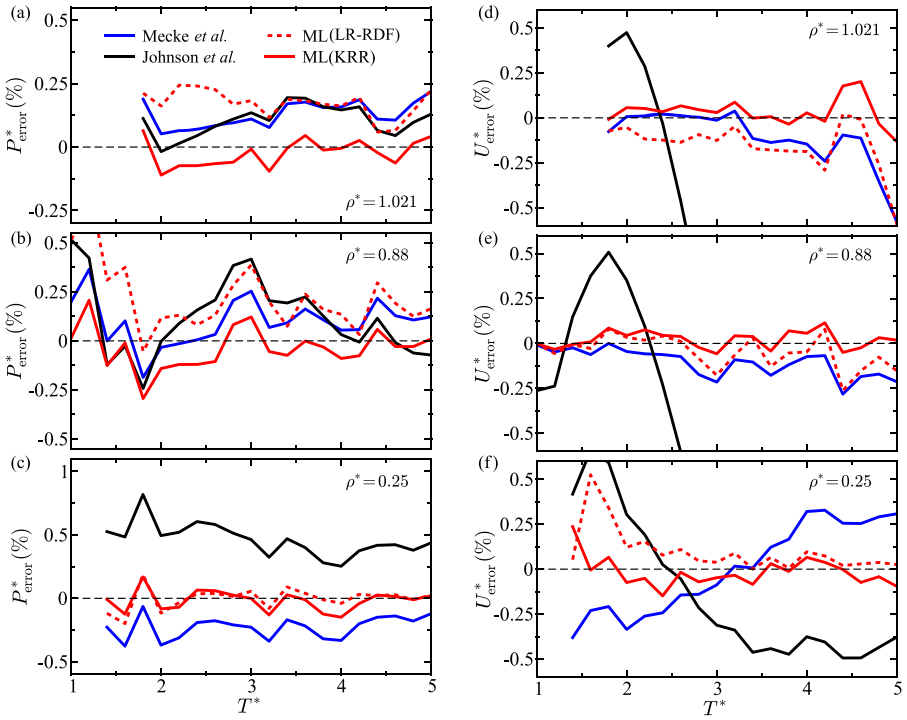

FIG. 8. Percent error in the pressure (left) and internal energy (right) calculated using the ML(KRR) (red) and ML(LR-RDF) (dashed red) methods and the expressions of Mecke *et al.* (blue) and Johnson *et al.* (black). The results are shown as a function of temperature along the isodensity contours $\rho^* = 1.021$ [(a) and (d)], $\rho^* = 0.88$ [(b) and (e)], and $\rho^* = 0.25$ [(c) and (f)]. The black dashed line in each panel denotes zero error.

the last column of the table. The Mecke expression is widely considered to be the most accurate LJ EOS for generating thermodynamic properties. The KRR process results in significant error decreases in comparison to the other methods, outperforming the Mecke EOS over every test set except the Johnson dataset where the methods yield similar results. Specifically, the KRR procedure reduces the error when compared to the Mecke EOS over the combined test set by a factor of ≈1.7. The LR-RDF method is approximately as accurate as the Johnson EOS and, in general, results in errors in the pressure that are between 2 and 10 times greater than that produced by the KRR method and the Mecke EOS. Because the KRR process is more accurate than the Mecke EOS, we conclude that the KRR process is currently the most accurate method for generating the pressure of the LJ system. The nearest neighbor approximation leads to significant errors, typically close to an order of magnitude greater than the other methods.

The average absolute percent error in the internal energy $\langle|U^*_{\text{error}}|\rangle$ generated by each theoretical method calculated over each test set is shown in Table III. In general, the LR-RDF method is more accurate than the Johnson EOS, but not as accurate as the Mecke EOS, while the KRR method reduces the absolute error in comparison to the Mecke EOS. Specifically, the developed KRR approach outperforms the Mecke EOS on every test set except for Johnson *et al.* data. When the absolute error is averaged over every point in all of the test sets combined, as shown in the last column of the table, the KRR method again reduces the error in the comparison to the Mecke EOS by roughly a factor of 1.7. We therefore conclude that the KRR approach developed here is currently

TABLE II. Average absolute percent error in the pressure $\langle|P^*_{\text{error}}|\rangle$ generated by several theoretical methods calculated over various test sets. The lowest value for each test set is shown in bold.

	Test data				
EOS/method	This work (265 pts)	Meier (269 pts)	Johnson *et al.* (134 pts)	Stephan *et al.* (317 pts)	Combined
Mecke *et al.*	0.151	0.350	**0.285**	0.338	0.284
Johnson *et al.*	0.305	0.632	0.536	0.770	0.575
ML(LR-RDF)	0.166	0.614	0.572	0.908	0.582
ML(KRR)	**0.089**	**0.270**	0.304	**0.079**	**0.164**
Nearest neighbor	1.768	1.845	0.409	6.191	3.027

TABLE III. Average absolute percent error in the internal energy $\langle|U^*_{\text{error}}|\rangle$ generated by several theoretical methods calculated using various test sets. The lowest value for each test set is shown in bold.

	Test data				
EOS/method	This work (265 pts)	Meier (269 pts)	Johnson *et al.* (134 pts)	Stephan *et al.* (317 pts)	Combined
Mecke *et al.*	0.132	0.177	**0.193**	0.229	0.184
Johnson *et al.*	0.813	0.608	1.082	0.468	0.683
ML(LR-RDF)	0.105	0.264	0.378	0.415	0.285
ML(KRR)	**0.066**	**0.104**	0.272	**0.080**	**0.109**
Nearest neighbor	0.905	1.758	0.431	6.088	2.741

the most accurate theoretical method for determining the internal energy of the LJ system. The nearest neighbor approximation is overall the worst method; however, over the Johnson *et al.* test data, this method generates more accurate predictions for the pressure than the LR-RDF method and the Johnson EOS and more accurate predictions for the internal energy than the Johnson EOS. This behavior is explained, in part, because the Johnson dataset is the smallest we have examined, and the values in this dataset are sampled from the fewest number of configurations in comparison to the other datasets, leading to uncertainty in the error calculations.

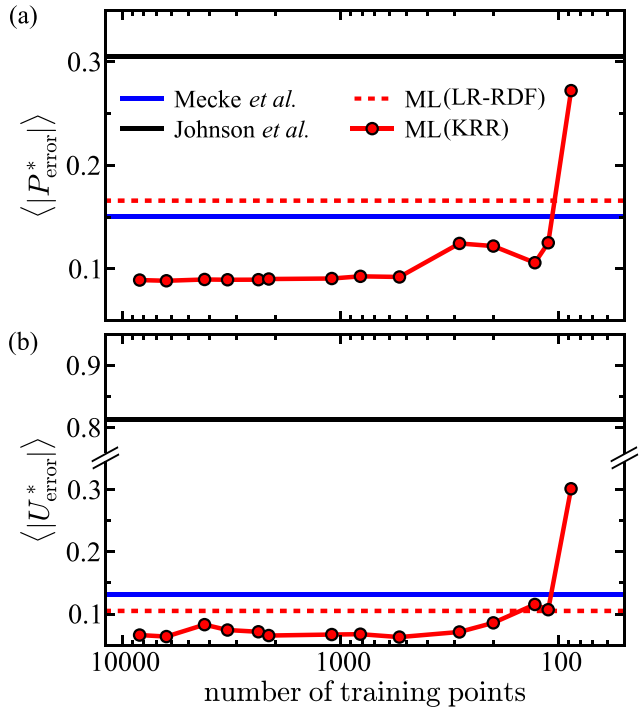

FIG. 9. Learning curve for the KRR process showing the average absolute percent error in the pressure (a) and internal energy (b) as a function of the number of training points. Results are shown for the KRR process (red), the EOS expressions of Mecke *et al.* (blue) and Johnson *et al.* (black), and the LR-RDF method (dashed red) trained using the data shown in Fig. 1. The number of training points is shown on a log scale.

Figure 9 shows learning curves for the KRR process. Specifically, the average absolute percent error in the pressure and internal energy are shown, respectively, in Figs. 9(a) and 9(b) as a function of the number of state points in the training set. The averages are taken over the test set constructed in this manuscript (see Sec. II C). For both of the thermodynamic properties, there are only small increases in the error as the training set is reduced in size from ~10 000 points to ~100 points. It is interesting to note that when the KRR process is trained on as few as ≈200 points, it provides more accurate predictions for both the pressure and internal energy in comparison to all of the other methods. The fluctuations in the learning curves arise because the size of the training dataset cannot be reduced in a completely uniform way because some regions of the phase diagram contain a higher density of state points than others in the full Gottschalk training set.

The EOS expressions that have been developed for the LJ system are some of the most accurate due to the importance of the LJ system as the prototypical fluid model and the simplicity of the interatomic LJ potential. It is significant that the ML approaches developed here are more accurate than previous methods/expressions applied to the LJ system, and we therefore expect that in other condensed-phase systems, ML will provide even larger increases in predictive accuracy than those observed here for the LJ fluid. For example, in complex fluid systems, analytical EOS expressions typically do not exist or they provide only a qualitative description of the system's thermodynamical properties. An advantage of the present ML approaches is that because no assumptions are made about the functional form of the EOS or the RDF, the developed methods are easily transferable to more complex systems. This opens the possibility to apply similar approaches to complex condensed-phase systems that have previously been intractable to treat with traditional methods other than brute-force simulation.

IV. CONCLUSIONS

Machine learning protocols have been developed that are able to determine structural and thermodynamic properties of a LJ fluid more accurately in comparison to previous human-guided fitting attempts and integral equation methods. This work

illustrates an example case in which ML methods can be used to supersede human efforts in the development of predictive functions that are generated by fitting large datasets. All of the raw training data used in this article as well as the ML programs that are used to compute the RDF, pressure, and internal energy of a LJ fluid have been made publicly available at https://github.com/gtcraven. The primary use of these programs is to avoid the computationally taxing task of performing molecular simulations in order to generate properties of the LJ system. This will be especially useful when a large number of simulations are needed over various parameter values for: interfacing between scales in multi-scale physics codes, analytical function fitting, molecular design, and other applications in which large batches of data are needed over diverse system states.

Applying the developed LR-RDF method was shown to dramatically improve the predictive accuracy of generating the RDF of a LJ fluid when compared to solving integral equations and fitting human-guided functional forms. Typically, we found that using the LR-RDF method reduced the error by an order of magnitude in comparison to traditional theoretical methods. The thermodynamic properties generated using the developed KRR ML method were also able to reduce the predictive accuracy when compared to previous methods, but by a smaller factor, typically of the order 2. The reason that the LR-RDF approach produces a much larger error decrease for structural properties than the KRR approach does for thermodynamic properties is because the dimensionality and complexity of the structural functions is greater, and therefore, previous theoretical methods have historically had difficulty producing accurate results for structural properties of the LJ fluid. Said in more detail, the LJ thermodynamic properties are fit to two-dimensional inputs, and therefore optimization and regression over this data is much simpler than performing regression using the three-dimensional inputs that are used to fit the LJ RDF.

The major disadvantage of applying the presented ML methods is that generating enough training data to sufficiently sample a particular region of a system's phase diagram can be computational expensive. Moreover, in some systems, this step may be so computationally taxing that it precludes the use of ML to develop predictive models. In fact, it should be noted that in any situation in which (a) data are only needed at a single state point, (b) a simulation methodology exists that can accurately produce properties of the system, and (c) no ML procedure has been previously trained to produce the properties of interest for the system, performing a molecular simulation at this single point will be much more computationally efficient than first generating training data and then implementing ML. As described in detail previously, however, we anticipate that there will be a number of systems and situations in which the present ML approaches will be useful.

We have applied ML to produce particular properties of the LJ fluid; however, the developed protocols should serve as general blueprints that can be used to predict other structural, thermodynamic, transport, and dynamical properties in a multitude of fluid systems, including soft and complex fluids.[66–75] Our future work will address applications of ML in the development of predictive functions for other properties of isotropic fluids. Extending the ML methods developed in this article to the case of anisotropic fluids is an important next step, and work in this direction is currently underway.

ACKNOWLEDGMENTS

We acknowledge support from the Los Alamos National Laboratory (LANL) Directed Research and Development funds (LDRD). This research was performed, in part, at the Center for Nonlinear Studies (CNLS) and the Center for Integrated Nanotechnologies (CINT) at LANL. The computing resources used to perform this research were provided by the LANL Institutional Computing Program. We thank Ryan Jadrich and David Rosenberger for insightful discussions.

DATA AVAILABILITY

The data and computational codes that support the findings of this study are openly available at https://github.com/gtcraven, Refs. 76 and 77.

REFERENCES

[1] M. P. Allen and D. J. Tildesley, *Computer Simulations of Liquids* (Oxford University Press, New York, 1987).
[2] J. P. Hansen and I. R. McDonald, *Theory of Simple Liquids* (Academic Press, San Diego, 1986).
[3] B. J. Alder, S. P. Frankel, and V. A. Lewinson, J. Chem. Phys. **23**, 417 (1955).
[4] B. J. Alder and T. E. Wainwright, J. Chem. Phys. **31**, 459 (1959).
[5] F. H. Stillinger and A. Rahman, J. Chem. Phys. **60**, 1545 (1974).
[6] K. Kremer and G. S. Grest, J. Chem. Phys. **92**, 5057 (1990).
[7] Y. Wang and G. A. Voth, J. Am. Chem. Soc. **127**, 12192 (2005).
[8] J. Jung, W. Nishima, M. Daniels, G. Bascom, C. Kobayashi, A. Adedoyin, M. Wall, A. Lappala, D. Phillips, W. Fischer *et al.*, J. Comput. Chem. **40**, 1919 (2019).
[9] R. Car and M. Parrinello, Phys. Rev. Lett. **55**, 2471 (1985).
[10] R. Kosloff, J. Phys. Chem. **92**, 2087 (1988).
[11] G. Kresse and J. Hafner, Phys. Rev. B **47**, 558 (1993).
[12] S. Izvekov and G. A. Voth, J. Chem. Phys. **123**, 134105 (2005).
[13] M. G. Saunders and G. A. Voth, Annu. Rev. Biophys. **42**, 73 (2013).
[14] J. F. Dama, A. V. Sinitskiy, M. McCullagh, J. Weare, B. Roux, A. R. Dinner, and G. A. Voth, J. Chem. Theory Comput. **9**, 2466 (2013).
[15] B. F. E. Curchod and T. J. Martínez, Chem. Rev. **118**, 3305 (2018).
[16] K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev, and A. Walsh, Nature **559**, 547 (2018).
[17] G. Carleo and M. Troyer, Science **355**, 602 (2017).
[18] N. Lubbers, J. S. Smith, and K. Barros, J. Chem. Phys. **148**, 241715 (2018).
[19] A. Moradzadeh and N. R. Aluru, J. Phys. Chem. Lett. **10**, 1242 (2019).
[20] G. T. Craven, N. Lubbers, K. Barros, and S. Tretiak, J. Phys. Chem. Lett. **11**, 4372–4378 (2020).
[21] J. Carrasquilla and R. G. Melko, Nat. Phys. **13**, 431 (2017).
[22] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Nature **549**, 195 (2017).
[23] D.-L. Deng, X. Li, and S. Das Sarma, Phys. Rev. X **7**, 021021 (2017).
[24] M. Mitchell, Information **10**, 51 (2019).
[25] R. Jia and P. Liang, arXiv:1707.07328 (2017).
[26] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, arXiv:1707.08945 (2017).
[27] D. Hendrycks and T. G. Dietterich, arXiv:1807.01697 (2018).
[28] G. Goswami, N. Ratha, A. Agarwal, R. Singh, and M. Vatsa, in Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[29] A. Rosenfeld, R. Zemel, and J. K. Tsotsos, arXiv:1808.03305 (2018).
[30] J. G. Kirkwood, V. A. Lewinson, and B. J. Alder, J. Chem. Phys. **20**, 929 (1952).
[31] W. W. Wood and F. R. Parker, J. Chem. Phys. **27**, 720 (1957).
[32] J. L. Yarnell, M. J. Katz, R. G. Wenzel, and S. H. Koenig, Phys. Rev. A **7**, 2130 (1973).
[33] S. Goldman, J. Phys. Chem. **83**, 3033 (1979).
[34] E. Matteoli and G. A. Mansoori, J. Chem. Phys. **103**, 4672 (1995).
[35] A. Morsali, E. K. Goharshadi, G. Ali Mansoori, and M. Abbaspour, Chem. Phys. **310**, 11 (2005).
[36] M. Bamdad, S. Alavi, B. Najafi, and E. Keshavarzi, Chem. Phys. **325**, 554 (2006).
[37] J. S. Emampour, A. Morsali, M. Sarvghadi, G. R. Jafari, N. Beyzaie, and S. A. Beyramabadi, Phys. Chem. Liq. **50**, 187 (2012).
[38] J. J. Nicolas, K. E. Gubbins, W. B. Streett, and D. J. Tildesley, Mol. Phys. **37**, 1429 (1979).
[39] J. E. Straub, M. Borkovec, and B. J. Berne, J. Chem. Phys. **89**, 4833 (1988).
[40] B. Smit, J. Chem. Phys. **96**, 8639 (1992).
[41] J. K. Johnson, J. A. Zollweg, and K. E. Gubbins, Mol. Phys. **78**, 591 (1993).
[42] M. Mecke, A. Müller, J. Winkelmann, J. Vrabec, J. Fischer, R. Span, and W. Wagner, Int. J. Thermophys. **17**, 391 (1996).
[43] M. Thol, G. Rutkai, R. Span, J. Vrabec, and R. Lustig, Int. J. Thermophys. **36**, 25 (2015).
[44] M. Thol, G. Rutkai, A. Köster, R. Lustig, R. Span, and J. Vrabec, J. Phys. Chem. Ref. Data **45**, 023101 (2016).
[45] S. Pieprzyk, A. C. Brańka, S. Maćkowiak, and D. M. Heyes, J. Chem. Phys. **148**, 114505 (2018).
[46] A. A. Broyles, S. U. Chung, and H. L. Sahlin, J. Chem. Phys. **37**, 2462 (1962).
[47] F. Mandel, R. J. Bearman, and M. Y. Bearman, J. Chem. Phys. **52**, 3315 (1970).
[48] J. A. Barker and D. Henderson, Rev. Mod. Phys. **48**, 587 (1976).
[49] D. M. Duh and A. D. J. Haymet, J. Chem. Phys. **103**, 2625 (1995).
[50] D. M. Duh and D. Henderson, J. Chem. Phys. **104**, 6742 (1996).
[51] N. Choudhury and S. K. Ghosh, J. Chem. Phys. **116**, 8517 (2002).
[52] Q. Wang, D. J. Keffer, D. M. Nicholson, and J. B. Thomas, Phys. Rev. E **81**, 061204 (2010).
[53] T. Miyata and K. Tange, Chem. Phys. Lett. **700**, 88 (2018).
[54] S. Y. Mashayak and N. R. Aluru, J. Chem. Phys. **148**, 214102 (2018).
[55] A. Moradzadeh and N. R. Aluru, J. Phys. Chem. Lett. **10**, 7568 (2019).
[56] M. Gottschalk, AIP Adv. **9**, 125206 (2019).
[57] S. Stephan, M. Thol, J. Vrabec, and H. Hasse, J. Chem. Inf. Model. **59**, 4248 (2019).
[58] Y. Liu, W. Hong, and B. Cao, Energy **188**, 116091 (2019).
[59] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning* (The MIT Press, 2012).
[60] A. E. Hoerl and R. W. Kennard, Technometrics **12**, 55 (1970).
[61] P. Exterkate, P. J. F. Groenen, C. Heij, and D. van Dijk, Int. J. Forecasting **32**, 736 (2016).
[62] N. Goga, A. J. Rzepiela, A. H. de Vries, S. J. Marrink, and H. J. C. Berendsen, J. Chem. Theory Comput. **8**, 3637 (2012).
[63] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, J. Mach. Learn. Res. **12**, 2825 (2011).
[64] K. Meier, Ph.D. thesis, University of the Federal Armed Forces Hamburg, 2002.
[65] T. B. Martin, T. E. Gartner, R. L. Jones, C. R. Snyder, and A. Jayaraman, Macromolecules **51**, 2906 (2018).
[66] B. M. Mladek, D. Gottwald, G. Kahl, M. Neumann, and C. N. Likos, Phys. Rev. Lett. **96**, 045701 (2006).
[67] C. N. Likos, Soft Matter **2**, 478 (2006).
[68] D. Coslovich and A. Ikeda, Soft Matter **9**, 6786 (2013).
[69] G. T. Craven, A. V. Popov, and R. Hernandez, J. Chem. Phys. **138**, 244901 (2013).
[70] G. T. Craven, A. V. Popov, and R. Hernandez, Soft Matter **10**, 5350 (2014).
[71] G. T. Craven, A. V. Popov, and R. Hernandez, J. Phys. Chem. B **118**, 14092 (2014).
[72] G. T. Craven, A. V. Popov, and R. Hernandez, J. Chem. Phys. **142**, 154906 (2015).
[73] N. B. Wilding and P. Sollich, J. Chem. Phys. **141**, 094903 (2014).
[74] S. Prestipino and F. Saija, J. Chem. Phys. **141**, 184502 (2014).
[75] R. S. Singh and R. Hernandez, Chem. Phys. Lett. **708**, 233 (2018).
[76] G. T. Craven, Machine learning codes for structural and thermodynamic properties of a Lennard-Jones fluid, 2020, URL: https://github.com/gtcraven/MachineLearning_LennardJones.
[77] G. T. Craven, Lennard-Jones radial distribution function dataset, 2020, URL: https://github.com/gtcraven/MachineLearning_LennardJones.