Machine learning of consistent thermodynamic models using automatic differentiation

David Rosenberger,[*] Kipton Barros, and Timothy C. Germann
Los Alamos National Laboratory, Theoretical Division, Physics and Chemistry of Materials Group, Los Alamos, New Mexico 87545, USA

Nicholas Lubbers[†]
Los Alamos National Laboratory, Computer, Computational & Statistical Sciences Division,
Information Sciences Group, Los Alamos, New Mexico 87545, USA

(Received 1 September 2021; accepted 9 March 2022; published 1 April 2022)

We propose a data-driven method to describe consistent equations of state (EOS) for arbitrary systems. Complex EOS are traditionally obtained by fitting suitable analytical expressions to thermophysical data. A key aspect of EOS is that the relationships between state variables are given by derivatives of the system free energy. In this work, we model the free energy with an artificial neural network and utilize automatic differentiation to directly learn the derivatives of the free energy. We demonstrate this approach on two different systems, the analytic van der Waals EOS and published data for the Lennard-Jones fluid, and we show that it is advantageous over direct learning of thermodynamic properties (i.e., not as derivatives of the free energy but as independent properties), in terms of both accuracy and the exact preservation of the Maxwell relations. Furthermore, the method implicitly provides the free energy of a system without explicit integration.

DOI: 10.1103/PhysRevE.105.045301

I. INTRODUCTION

Thermophysical properties of liquids are routinely measured in laboratory and computational experiments and then summarized in corresponding data sets [1–4]. However, the amount of available data is small compared to the total number of systems of interest for both pure components and mixtures [5]. Given the time and resource cost of direct measurements, scientists have used empirical functional forms for equations of state (EOS), which model the relationship of state variables like density, pressure, and temperature [6–8]. The derivation of those functional forms has been driven by a combination of physical laws, statistical mechanics, and choice of empirical functional forms [9,10]. Although these approaches have proven useful, limitations in applicability of statistical mechanics for complex systems and in choice of functional form may hinder the development of a general ansatz to estimate EOS. Therefore, we propose to utilize automatic differentiation (AD) [11,12] in combination with artificial neural networks (ANNs) [13] to model thermophysical properties and to implicitly compute the free energy of a system. AD consists of a set of high-level algorithms for efficiently computing certain derivatives of computations without numerical approximation. The goal of the proposed method is to learn the properties of interest as a derivative of a model free energy with respect to density, pressure, and temperature, thereby ensuring the thermodynamic consistency of the resulting model. This process can be applied regardless of the availability of free energy data in the examined data set.

A similar philosophy has been utilized by Christensen *et al.* to learn potential energy surfaces for atoms using response operators [14]. Rather than kernel-based machine learning (ML) methods applied by Christensen *et al.*, we employ neural networks, which is advantageous because the response functions are automatically generated.

ML approaches to fitting EOS have recently been studied [15,16]. EOS fits are often achieved by direct minimization of the difference between a measured property and its corresponding predicted value [17,18]. Although straightforward, this direct approach can lead to inconsistencies in the EOS via violation of the Maxwell relations, as we will show. Another approach to model EOS is to train ML models to the function that, when differentiated, generates the quantities of interest. Those functions can either be particle-particle interactions [19], the partition function [20], or the system's free energy [21]. However, those methods rely on accurate sampling of the phase space, which for most systems is a nontrivial task. Therefore, we propose an indirect method to model the free energy of a system from data on the free energy derivatives, which are easier to compute and thus effectively reduce the number of expensive and complex computations of the phase space. The power of neural networks to estimate a free energy has been demonstrated by Nicoli *et al.* in the context of lattice fluid theory [22]. The authors train an estimator in order to predict the free energy from data sets of microscopic configurations. This is conceptually different from the method proposed here, which learns from thermodynamic observables that can be, for example, measured in experiment. We suggest training a NN in order to predict derivatives of the free energy and by doing so implicitly matching the underlying free energy.

[*] Present address: Freie Universität Berlin, Germany; d.rosenberger@fu-berlin.de

[†] nlubbers@lanl.gov

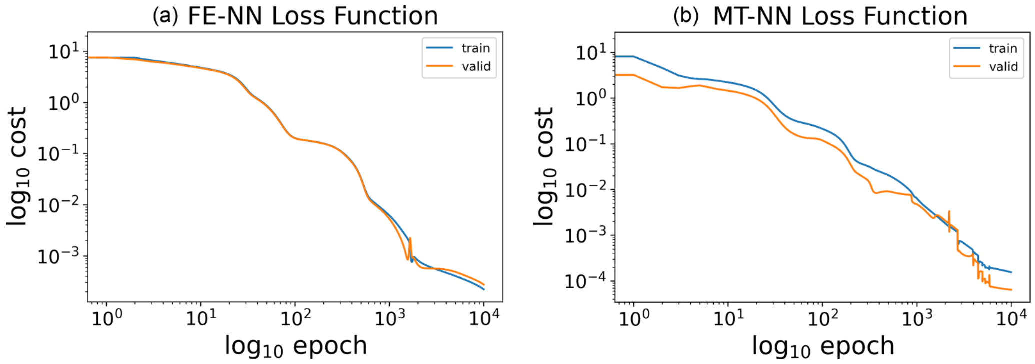

FIG. 1. (a) Loss function for the FE-NN model. Blue corresponds to the training set and orange to the validation set. (b) Loss function for the MT-NN model. Blue corresponds to the training set and orange to the validation set.

II. METHOD

While the model could be based in any thermodynamic potential, in this work we choose to start with the Helmholtz free energy A defined in the canonical ensemble:

$$A(N, V, T) = U - TS. \tag{1}$$

A is a function of the particle number N, the volume V, and temperature T. U is the internal energy of a system, and S is the entropy. Thermophysical properties such as pressure P, chemical potential μ, and the entropy S can be computed from first-order partial derivatives of the free energy: $P = -(\frac{\partial A}{\partial V})_{N,T}$, $\mu = (\frac{\partial A}{\partial N})_{V,T}$, $S = -(\frac{\partial A}{\partial T})_{V,T}$, $U = A + TS = A - T(\frac{\partial A}{\partial T})_{V,T}$. Further, properties like the isothermal compressibility β, the thermal pressure coefficient γ, the isochoric heat capacity c_v, and the thermal expansion coefficient α can be computed from first-order derivatives of the thermophysical properties P and U:

$$\beta = -\frac{1}{V}\left(\frac{\partial V}{\partial P}\right)_{N,T} = \frac{1}{V}\left(\frac{\partial\left(\frac{\partial A}{\partial V}\right)_{N,T}}{\partial V}\right)_{N,T}^{-1}, \tag{2}$$

$$\gamma = \left(\frac{\partial P}{\partial T}\right)_{V,T} = \left(\frac{-\partial\left(\frac{\partial A}{\partial V}\right)_{N,T}}{\partial T}\right)_{V,T}, \tag{3}$$

$$c_v = \left(\frac{\partial U}{\partial T}\right)_{V,T} = \left(\frac{\partial\left[A - T\left(\frac{\partial A}{\partial T}\right)_{V,T}\right]}{\partial T}\right)_{V,T}, \tag{4}$$

$$\alpha = \beta\gamma. \tag{5}$$

While these relations are well known, it is difficult to solve them in practice because either the functional form of the free energy is not known or it is too complicated to directly compute the free energy in order to apply numerical differentiation. Therefore, we propose that the free energy of a system $\hat{A}$ be described with a learned function $\hat{f}$ which depends on the density $\rho = N/V$ and the temperature T. $\hat{f}$ is expressed in form of an ANN, whose parameters Θ (weights and biases for each layer) are optimized during learning:

$$\hat{A} = N\hat{f}_\Theta[\rho, T]. \tag{6}$$

The factor N ensures that the learned free energy $\hat{A}$ is an extensive function, whereas the learning itself is performed on intensive variables.

From now on we refer to this model as the Free Energy Neural Network (FE-NN) model. The parameters Θ are optimized by minimization of the following loss function:

$$L[\Theta] = \lambda_P\left[-\left(\frac{\partial \hat{A}}{\partial V}\right)_{N,T} - P\right]^2 + \lambda_\mu\left[\left(\frac{\partial \hat{A}}{\partial N}\right)_{V,T} - \mu\right]^2 + \lambda_U\left[\hat{A} - T\left(\frac{\partial \hat{A}}{\partial T}\right)_{V,T} - U\right]^2, \tag{7}$$

where $\hat{A}$ is the learned free energy, and λ_P, λ_μ, and λ_U are weighting parameters whose exact forms are

$$\lambda_P = \frac{1}{\frac{1}{l}\sum_{i=1}^{l}(P_i - \langle P\rangle)^2} = \frac{1}{\sigma_P}, \tag{8}$$

$$\lambda_\mu = \frac{1}{\frac{1}{l}\sum_{i=1}^{l}(\mu_n - \langle \mu\rangle)^2} = \frac{1}{\sigma_\mu}, \tag{9}$$

$$\lambda_U = \frac{1}{N^2\sum_{i=1}^{l}\left(\frac{U_i - \langle U\rangle}{N_i}\right)^2} = \frac{1}{N^2\sigma_{\frac{U}{N}}}, \tag{10}$$

where l equals the number of points in a corresponding data set and all variables have the same meaning as previously defined. Each term of the loss function is normalized with the corresponding variance σ. We desire to ensure that all terms in the loss are intensive; therefore for the internal energy we have to take the particle number N into account. The loss function summarizes the main idea of this work: We are utilizing AD *within the model* to seek the best approximation for the free energy A subject to the known values of thermophysical properties P, μ, and U. This is in addition to the standard use of AD in computing the derivative of the loss function with respect to the parameters, $\nabla_\Theta L$, to update the network; multiple, nested calls to AD are made to achieve fitting to the derivatives of the free energy. Thus if we learn the derivatives of the free energy, we implicitly also model the free energy of the system without performing an explicit integration. At the same time, we can further utilize AD to compute second derivatives β, γ, and c_v. This enables a complete and consistent thermodynamic modeling technique for an arbitrary material from data. We contrast this with a typical ML approach, where differences between target and predicted

TABLE I. Mean absolute error between predicted and true data generated from the VdW EOS for the following: the chemical potential μ, pressure P, internal energy U, and isothermal compressibility β. For the FE-NN model, we also give the corresponding errors for the free energy A and entropy S.

Model	μ	P	U	A	S	β
FE-NN	0.0009	0.0003	0.0010	0.0005	0.0032	0.2331
MT-NN	0.0172	0.0023	0.0042	—	—	1.0270
KRR	0.0038	0.0077	0.0002	—	—	16.2914

properties are directly minimized. For this Multi-Task Neural Network (MT-NN) model the loss function takes the following form: $L[\Theta] = \lambda_P[\hat{P} - P]^2 + \lambda_\mu[\hat{\mu} - \mu]^2 + \lambda_U[\hat{U} - U]^2$, where $\hat{P}$, $\hat{U}$, and $\hat{\mu}$ indicate the properties learned by the network. The MT-NN uses the same weight functions λ and intensive learning formulation as the FE-NN; inputs are intensive, and the extensive energy is predicted with a scaling factor of N.

III. TRAINING PROCEDURE AND DATA PREPARATION

A. Van der Waals equation of state

The first test case for both ML models is the the van der Waals equation of state (VdW EOS) defined by its underlying free energy:

$$A(N, V, T) = -Nk_BT\left[1 + \ln\left(\frac{(V - Nb)T^{3/2}}{N\Phi}\right) - \frac{aN^2}{V}\right]. \tag{11}$$

For constants, $a = 0.01$, $b = 0.01$, and $\Phi = 1.0$, we generate 1000 random state points for each independent variable. For N, 1000 values are generated in the range between 0.5 and 2. For V, 1000 random values are generated in the range between 3×2.8 and $3/2.8$. This leads to reduced volumes $v = V/N$ in the range between $20\,v_c$ and $500\,v_c$, where v_c is the critical volume. Finally, for T 1000 values are randomly generated in the range between $(8/27)/3$ and $(8/27) \times 3$, leading to temperatures above and below the critical temperature of the VdW gas. This represents a physical but simple thermodynamic system of a low-density gas. The true values for the free energy are computed according to Eq. (11) for this data set. By applying automatic differentiation on that free energy the corresponding true values for pressure P, chemical potential μ, internal energy U, and entropy S are obtained. As stated above, learning of the FE-NN model is performed on a per-atom (intensive) basis. We feed $\rho = N/V$ and T to the neural network, and all extensive target properties are normalized with the corresponding particle number. The initial weights of the ANN are assigned with the Xavier function [23] with a gain of 1.5. Training is performed for 10 000 epochs with a batch size of 20. Each neuron is activated with the Softmax function [24,25]. The Adam optimizer is applied to minimize the loss function with an initial learning rate of 0.001. If the loss function does not change over 500 epochs, the learning rate is automatically decreased by a factor of 0.5. We train both the FE-NN model and MT-NN model, to 20% of the 3000 data points, use 10% for validation, and test on the remaining 70%.

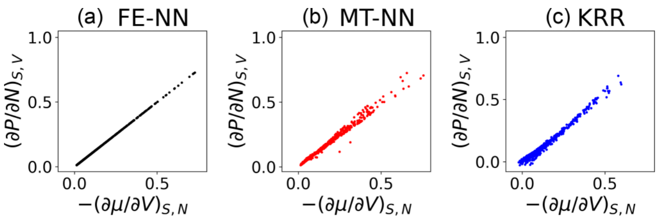

FIG. 2. Maxwell relation for the VdW EOS evaluated from (a) the FE-NN, (b) the MT-NN, and (c) the KRR model.

B. Lennard-Jones system

The data in [26] provide a suitable test case for the two ML approaches to estimate EOS. Thermodynamic properties have been collected from finite-size molecular dynamics simulation of the Lennard-Jones (LJ) fluid that covers homogeneous phases, vapor-liquid equilibrium, and regions above and below the critical point of the models. However, not all data sets summarized therein contain all thermophysical properties. Therefore, we first extract all data sets which contain the following thermophysical properties: μ, P, U, α, β, γ, and c_v. Second, some regions in phase space are sparsely sampled and are thus difficult to learn. As such, we include only points with temperature $T < 2$ and a density $\rho > 0.05$. All values are given in LJ units. In the end, we extract data for 1721 state points. In the data set only the excess part of properties like the chemical potential are provided; thus we manually added the ideal gas contribution where needed. Further, the data set contains only information on the density and not on particle numbers and volume. We therefore generated random particle numbers in the range between 0 and 1, and computed the volume according to $V = N/\rho$. Training is performed in total on 70% on the data total: 40% are used for direct training, and 30% are considered for validation. The test set contains the remaining 30%. The hyperparameters for learning are the same as for the VdW EOS.

IV. RESULTS

In the following, we show results from the subsequent network architecture for the free energy model: one input

TABLE II. Standard deviations (σ) for the mean absolute error values reported in Table I.

Model	σ_μ	σ_P	σ_U	σ_A	σ_S	σ_β
FE-NN	3.37×10^{-4}	5.8×10^{-5}	4.11×10^{-4}	3.79×10^{-4}	1.871×10^{-3}	8.38474×10^{-2}
MT-NN	1.83860×10^{-2}	2.5586×10^{-3}	5.2906×10^{-3}	—	—	1.100733
KRR	9.11×10^{-4}	1.3077×10^{-3}	0	—	—	9.84037

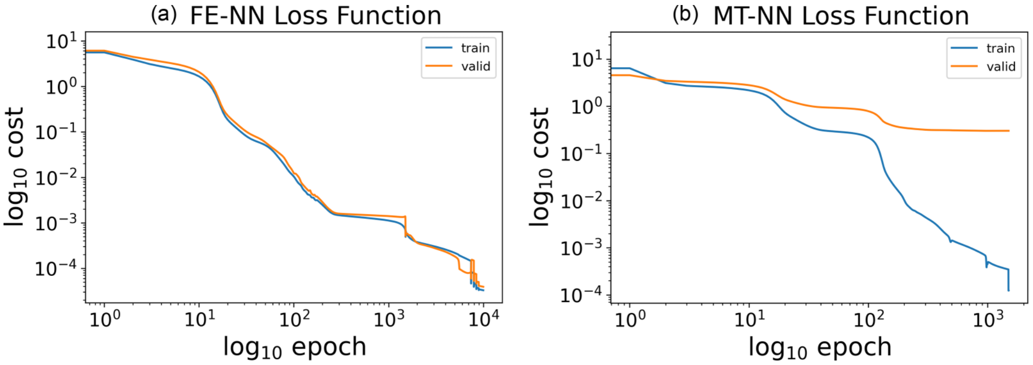

FIG. 3. (a) Loss function for the FE-NN model. Blue corresponds to the training set and orange to the validation set. (b) Loss function for the MT-NN model. Blue corresponds to the training set and orange to the validation set.

layer with two neurons, three hidden layers with 16 neurons, and one output layer with one neuron for free energy. The ANN for the multitask learner has the same internal structure, but one output layer each for $\hat{P}$, $\hat{U}$, and $\hat{\mu}$. Results for other architectures are presented in corresponding Tables S I–III and S V–VII in the Supplemental Material [27]. Our models are implemented in PyTorch [28].

A. Van der Waals equations of state

We first assess the two models on the the VdW EOS. This simple test case allows a comparison of the ML models with an analytical expression. In Fig. 1(a) we show the loss function of the FE-NN model as a function of the number of epochs. The blue line corresponds to the randomly selected 20% training data, whereas the orange line corresponds to the 10% of the training data which are used for validation. As one sees, the learning rate is lowered, which means a low-cost combination of parameters is found. Additionally, the final model chosen is the one that minimizes the validation error, also known as an early stopping procedure. In Fig. 1(b) we show the loss function of the MT-NN model, which shows similar behavior. In Table I we present the mean absolute error (MAE) values between the ground truth and the ML predictions. The MAE values are averaged over four ML runs, each initialized with a different random seed for the training, validation, and test splitting. For more details see Table S IV [27]. One sees that the FE-NN is consistently more accurate than the MT-NN. In addition we also present results for a kernel ridge regression (KRR) ML model for μ, P, and U. This model has been established in [18] to learn P and U for LJ fluids. The model utilizes a polynomial kernel of the fourth order:

$$K(x_i, x_j) = \left(\gamma x_i^T x_j + c_0\right)^4, \tag{12}$$

where x_i and x_j are input features. γ and c_0 are optimized using a grid search where the optimization scoring function was the mean absolute percent error of the corresponding thermodynamic property. The conditioning factor α [see Eq. (14)] in the KRR process is set to one. As one sees, with KRR one obtains MAE values which are intermediate to the NN models for the three target properties μ, P, and U. In Table II we show the standard deviation for the MAE values computed over the different random seeds used to split the data (see Table S IV for more information [27]). We see that on average the standard deviations for the FE-NN model are at least one order of magnitude smaller than for the MT-NN model. The KRR model is again intermediate between the two NN models. This confirms the higher accuracy of the FE-NN model over the MT-NN model indicated from the MAE values. Moreover, in Table I we present the MAE values for the free energy A and the entropy S using the FE-NN. These

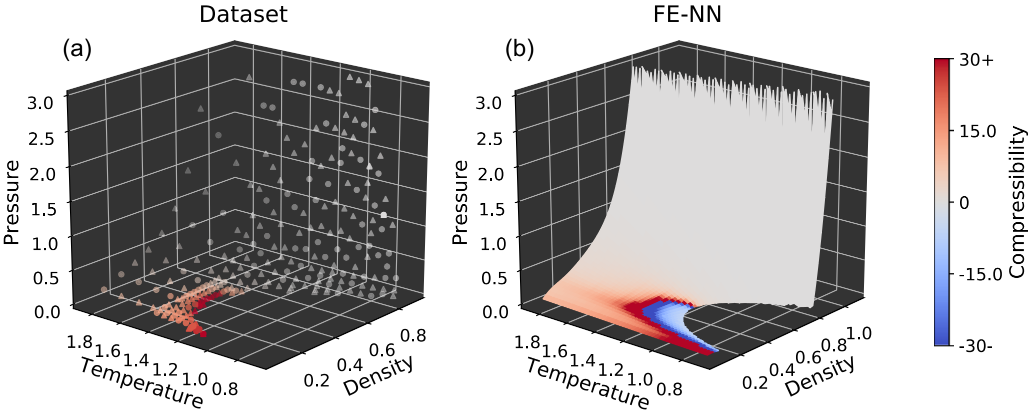

FIG. 4. The LJ system. Color indicates isothermal compressibility β. (a) Data; training points (circles) and validation and testing points (triangles). (b) FE-NN. Note that the model predicts an unstable region even when no data are available in this region of phase space.

quantities are completely unavailable in the MT-NN and KRR model, but easy to compute with the FE-NN using AD. Although A and S are not explicitly part of the training, the FE-NN achieves high accuracy for both properties, whereas the MT-NN and the KRR have no notion of these two quantities. Information regarding the free energy and entropy enter the training data only through their derivatives, and as a consequence MT-NN and KRR fail to infer prediction functions which can be consistently thermodynamically integrated, because the Maxwell relations are violated as we will show. Only the FE-NN model is able to implicitly learn the free energy of a system because it treats μ, P, and U as true derivatives of a single state function and thus produces predictions which can be consistently thermodynamically integrated. In addition, we assess the performance of the three models in predicting the isothermal compressibility. The analytical value for β for the VdW EOS is

$$\beta = -\frac{1}{V}\frac{1}{\left(\frac{\partial P}{\partial V}\right)_T} = -\frac{1}{V}\frac{1}{\frac{2aN^2}{V^3} - \frac{NkT}{(V-Nb)}}. \tag{13}$$

The values for the ML models are obtained from differentiation of the learned pressure with respect to the volume [see Eq. (2)]. For the NN models this is easily be obtained via AD from the network architecture, whereas for KRR we differentiate via the equation

$$y_{\text{pred}}(x) = K(x, x_i)\alpha = \left(\gamma x_i^T x + c_0\right)^4 \alpha, \tag{14}$$

$$\frac{\partial y_{\text{pred}}(x)}{\partial q} = 4\gamma x_j (\gamma x^T x_j + c_0)^3 \alpha \frac{\partial x}{\partial q}, \tag{15}$$

where y_{pred} is a predicted quantity (μ, P, or U) from the Kernel model, q is a thermodynamic input (N, V, or T), and all other variables are as previously defined. From the MAE of β (see Table I) one sees that results for the FE-NN are more accurate than for the MT-NN for which the MAE is about five times larger. The KRR model has the largest MAE value for β. Besides the higher accuracy in reproducing thermophysical properties, only the FE-NN is able to construct a consistent thermodynamic model. To show this, we validate both models against the Maxwell relation:

$$\left(\frac{\partial P}{\partial N}\right)_{S,V} = -\left(\frac{\partial \mu}{\partial V}\right)_{S,N}. \tag{16}$$

The derivatives for the NN model are again obtained via AD and for KRR via Eq. (15). From Fig. 2 one sees that only the FE-NN can exactly recover the Maxwell relation, whereas the other two models show deviations from exact correlation. This demonstrates explicitly that there is no free energy state function that corresponds to the predictions of the MT-NN and the KRR approach, highlighting the key advantage of the FE-NN approach. In light of the fact that the this dilute VdW system is so simple, this shows that there is a fundamental inconsistency that arises when applying ML to thermodynamic data without respecting the Maxwell relations, which can be overcome with the FE-NN.

B. Lennard-Jones equations of state

In order to show the generality of the method, we now move to the more applied example, EOS for LJ fluids [29]. In Fig. 3(a) we show the loss function of the FE-NN model as a function of the number of epochs. The blue line corresponds to the randomly selected 40% of the data which has been used for training, whereas the orange line corresponds to left-out 30% of the training data set against which the model is validated. We can see that for both data sets the loss function is well optimized. In Fig. 3(b) we show the loss function of the MT-NN model as a function of the number of epochs. The MT-NN learner shows acceptable minimization of the loss over the training data. However, for validation it reaches a plateau value, which indicates overfitting and consequently will lead to poor transferability. We want to emphasize that overfitting does not occur in general for the MT-NN model. We tested different network architectures (see Tables S V–VII [27]) and different random seeds (Table S VIII [27]) to split the data into training, validation, and test, and only in some cases is the overfitting observed. In Fig. 4(a) we illustrate some of the phase space covered. Figure 4(b) shows the FE-NN reconstruction in this space. In Fig. 5 we validate both methods against the Maxwell relation of Eq. (16), and see again that only the FE-NN can accurately recover it, just as in the VdW system. Table III presents the MAE values for both models for first- and second-order thermodynamic properties on the LJ system. The FE-NN again does consistently better than the MT-NN, especially for the second-order derivatives. MAE values are averaged over four independent runs using different seed values for training, validation, and test splitting (see Table S VIII [27]), the corresponding standard deviations are reported in Table IV. The large deviations for β and c_v were found to be associated with the critical region [see phase diagrams in Figs. S4 (A)–(C) and (H)–(I) [27]]. This highlights possible grounds for improvement in the model; smooth models can encounter difficulties in the presence of first- or second-order phase transitions. Finally, we present a practical demonstration of the model in Fig. 6. The FE-NN was applied to a 1000 × 1000 grid of points covering the full range of

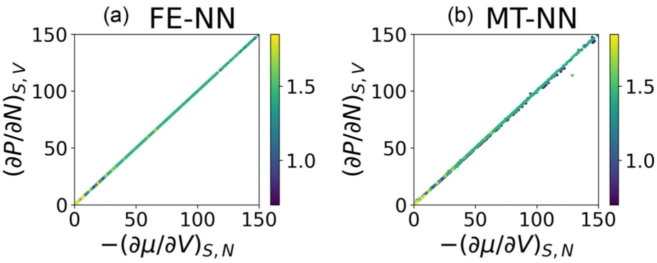

FIG. 5. Maxwell relation for the LJ EOS evaluated from (a) the FE-NN and (b) the MT-NN.

TABLE III. Mean absolute error between predicted and true data obtained from the LJ EOS [26]. Thermophysical properties considered are the isothermal compressibility β, thermal pressure coefficient γ, isochoric heat capacity c_v, and thermal expansion coefficient α.

Model	μ	P	U	β	γ	c_v	α
FE-NN	0.051	0.028	0.010	3.623	0.144	0.131	0.613
MT-NN	0.112	0.064	0.038	14.175	0.648	0.145	7.174

TABLE IV. Standard deviations (σ) for the mean absolute error values reported in Table III.

Model	σ_μ	σ_P	σ_U	σ_β	σ_γ	σ_{c_v}	σ_α
FE-NN	1.74×10^{-2}	1.78×10^{-2}	5.6×10^{-3}	1.4631	7.22×10^{-2}	6.40×10^{-2}	2.932×10^{-1}
MT-NN	6.08×10^{-2}	3.26×10^{-2}	2.69×10^{-2}	8.8227	4.455×10^{-1}	7.00×10^{-2}	3.5797

temperatures and densities present in the LJ data set. This process is extremely fast, taking less than 10 s on standard laptop CPU. We then used the Maxwell construction numerically to define the pressure as a function of density through the coexistence region for each isotherm. We plot the density and specific internal energy with respect to the pressure and temperature in the vicinity of the critical point. The locations where the Maxwell construction was applied are plotted as a dashed line ending in a star; we emphasize that this line was constructed using the model itself. Immediately visible in the plot is the classic structure of the first-order liquid-gas phase transition and its associated critical point. This recapitulates the classic textbook picture of a low-density gas comprising unbound particles, a high-density liquid comprising bound particles, and a continuous supercritical region that connects them. The plots are also smooth, showing that the outputs of the neural network are well behaved and do not contain any significant high-frequency artifacts.

V. CONCLUSION

In conclusion, we compared a standard Multi-Task model (MT-NN) to a Free Energy Neural Network (FE-NN) trained to predict free energy implicitly using automatic differentiation. We showed that only the FE-NN model is able to yield a consistent thermodynamic model that obeys the Maxwell relations. Further, it more accurately learns all thermodynamic properties for both test cases, the VdW and the LJ data sets. Additionally, for the VdW EOS we indeed provided evidence that the FE-NN implicitly learns the free energy and entropy accurately without including them explicitly in the training of the model. Because the Maxwell relations are satisfied, the learned free energy model is suitable for the computation of thermodynamic cycles or the embedding of the model in a larger simulation context (e.g., hydrodynamic or kinetic evolution). This implicit learning of free energies may as well be useful beyond learning EOS. The free energy is a fundamental property in biophysical systems [30] and for the development of coarse-grained models [31,32]. However, to be applicable for those more complex problems further improvements may be desired. To improve the behavior near a phase transition, it is possible to add another layer of automatic differentiation during training in order to include second-order partial derivatives in the model construction. Or if the analytic structure of the phase transition is known, future work may allow one to build this structure into the FE-NN, by designing a singularity in the activation functions that corresponds to the universality class of the transition. Generalizing the FE-NN technique to different thermodynamic potentials and to multicomponent systems is straightforward, and the analytical advantages will undoubtedly transfer, but the empirical error of such models has yet to be established. Whereas all these approaches suggest further improvements, the presented FE-NN model is the first step to elegantly learn thermodynamic properties and their generating free energy, without requiring the simulations to measure the free energy itself, and in principle without the need of advanced or enhanced simulation techniques [33–36]; simulations to measure thermodynamic responses (free energy derivatives) are often more straightforward, can be performed in the most convenient ensemble, and are easily parallelized across the thermodynamic phase space.

ACKNOWLEDGMENTS

The authors thank Mohamed Mehana, David Sanchez, and Galen Craven for fruitful discussions. This work was supported by the Laboratory Directed Research and Development program of Los Alamos National Laboratory (LANL) under Project No. 20190005DR. LANL is operated by Triad

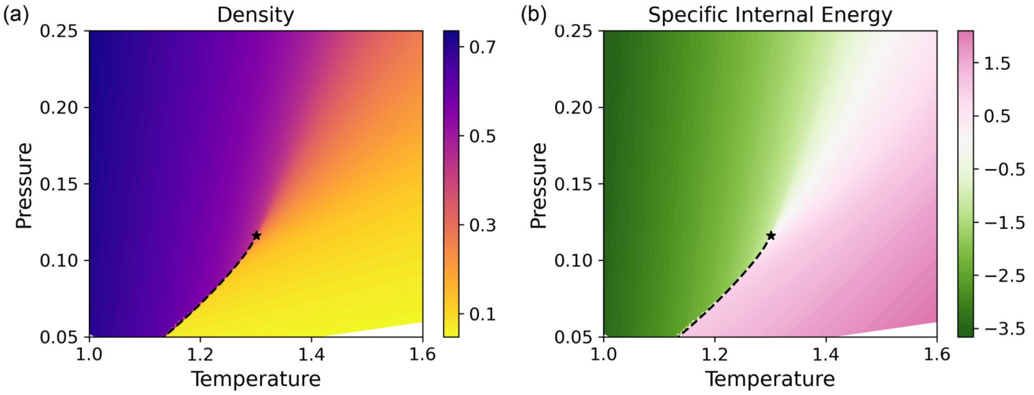

FIG. 6. Results of a FE-NN for LJ data as a function of pressure and temperature. (a) Density (N/V); (b) specific internal energy (U/N). The dashed line indicates where the Maxwell construction was required, corresponding to the first-order phase transition. The star shows the highest temperature value for the dashed line, corresponding to the model's critical point.

National Security, LLC, for the National Nuclear Security Administration of U.S. Department of Energy (Contract No. 89233218CNA000001). D.R. acknowledges partly funding from the Center for Nonlinear Studies (CNLS) at LANL. K.B. acknowledges support from the Center of Materials Theory as a part of the Computational Materials Science (CMS) program, funded by the U.S. Department of Energy, Office of Basic Energy Sciences.

[1] B. E. Gammon, *DIPPR 882— Properties and Related Thermodynamic Data of Binary Mixtures, Parts 1–4* (Design Institute for Physical Property Research/AIChE, 1997).

[2] P. Linstrom, NIST Chemistry WebBook, NIST Standard Reference Database 69, https://webbook.nist.gov/chemistry/ (1997).

[3] J. Marrero, J. Abildskov, S. Kalakul, and M. D. Ellegård, *Solubility and Related Properties of Large Complex Chemicals*, Chemistry Data Series Vol. 15 (DECHEMA, Frankfurt am Main, 2003).

[4] Royal Chemical Society, ChemSpider, https://www.chemspider.com (2021).

[5] L. Ruddigkeit, R. van Deursen, L. C. Blum, and J.-L. Reymond, J. Chem. Inf. Model. **52**, 2864 (2012).

[6] I. Müller, *A History of Thermodynamics: The Doctrine of Energy and Entropy* (Springer-Verlag, Berlin, 2007).

[7] J. S. Rowlinson, *Cohesion: A Scientific History of Intermolecular Forces* (Cambridge University Press, Cambridge, 2005).

[8] P. Ungerer, C. Nieto-Draghi, B. Rousseau, G. Ahunbay, and V. Lachet, J. Mol. Liq. **134**, 71 (2007).

[9] G. M. Kontogeorgis and G. K. Folas, *Thermodynamic Models for Industrial Applications: From Classical and Advanced Mixing Rules to Association Theories* (Wiley, Chichester, 2010).

[10] C. Nieto-Draghi, G. Fayet, B. Creton, X. Rozanska, P. Rotureau, J.-C. de Hemptinne, P. Ungerer, B. Rousseau, and C. Adamo, Chem. Rev. **115**, 13093 (2015).

[11] A. Griewank, in *Mathematical Programming—Recent Developments and Applications* (Kluwer Academic Publishers, Amsterdam, 1989), pp. 83–108.

[12] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, J. Mach. Learn. Res. **18**, 1 (2018).

[13] Y. LeCun, Y. Bengio, and G. Hinton, Nature (London) **521**, 436 (2015).

[14] A. S. Christensen, F. A. Faber, and O. A. von Lilienfeld, J. Chem. Phys. **150**, 064105 (2019).

[15] K. Zhu and E. A. Müller, J. Phys. Chem. B **124**, 8628 (2020).

[16] C. Dai and S. C. Glotzer, J. Phys. Chem. B **124**, 1275 (2020).

[17] Y. Liu, W. Hong, and B. Cao, Energy **188**, 116091 (2019).

[18] G. T. Craven, N. Lubbers, K. Barros, and S. Tretiak, J. Chem. Phys. **153**, 104502 (2020).

[19] F. Berressem and A. Nikoubashman, J. Chem. Phys. **154**, 124123 (2021).

[20] C. Desgranges and J. Delhommelle, J. Phys. Chem. C **124**, 1907 (2020).

[21] D. Wu, L. Wang, and P. Zhang, Phys. Rev. Lett. **122**, 080602 (2019).

[22] K. A. Nicoli, C. J. Anders, L. Funcke, T. Hartung, K. Jansen, P. Kessel, S. Nakajima, and P. Stornati, Phys. Rev. Lett. **126**, 032001 (2021).

[23] X. Glorot and Y. Bengio, in *Proc. of the 13th Intl. Conf. on Artificial Intelligence and Statistics, Proceedings of Machine Learning Research*, Vol. 9, edited by Y. W. Teh and M. Titterington (PMLR, 2010), pp. 249–256.

[24] R. D. Luce, *Individual Choice Behavior: A Theoretical Analysis* (Dover Publications, Mineola, NY, 2005).

[25] J. S. Bridle, in *NIPS'89: Proc. of the 2nd Intl. Conf. on Neural Information Processing Systems*, edited by D. S. Touretzky (MIT Press, MA, United States, 1989), p. 211.

[26] S. Stephan, M. Thol, J. Vrabec, and H. Hasse, J. Chem. Inf. Model. **59**, 4248 (2019).

[27] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevE.105.045301 for results on different network architectures and different data splits as well as validation of the automatic differentiation for second-order derivatives and correlation plots and corresponding phase diagrams for the Lennard-Jones system.

[28] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, *PyTorch: An Imperative Style, High-Performance Deep Learning Library* (Curran Associates Inc., Red Hook, NY, USA, 2019).

[29] M. Thol, G. Rutkai, A. Köster, R. Lustig, R. Span, and J. Vrabec, J. Phys. Chem. Ref. Data **45**, 023101 (2016).

[30] M. Gruebele, Curr. Opin. Struct. Biol. **12**, 161 (2002).

[31] W. G. Noid, J. Chem. Phys. **139**, 090901 (2013).

[32] M. S. Shell, in *Advances in Chemical Physics*, edited by S. A. Rice and A. R. Dinner (John Wiley & Sons, Hoboken, NJ, 2016), pp. 395–441.

[33] W. F. van Gunsteren and H. J. C. Berendsen, J. Comput.-Aided Mol. Des. **1**, 171 (1987).

[34] F. Wang and D. P. Landau, Phys. Rev. Lett. **86**, 2050 (2001).

[35] C. Abrams and G. Bussi, Entropy **16**, 163 (2013).

[36] F. Noé, S. Olsson, J. Köhler, and H. Wu, Science **365**, eaaw1147 (2019).