Bridge function as a functional of the radial distribution function: Operator learning and application

Martin Panholzer [*]
Uni Software Plus GmbH, Linzer Strasse 6, 4320 Perg, Austria

Michael Haring and Thomas Wallek [†]
Institute of Chemical Engineering and Environmental Technology, Graz University of Technology, Inffeldgasse 25/C, 8010 Graz, Austria

Robert E. Zillich [‡]
Institute for Theoretical Physics, Johannes Kepler University Linz, Altenberger Straße 69, 4040 Linz, Austria

(Received 26 May 2025; accepted 12 August 2025; published 15 September 2025)

Properties of classical molecular systems can be calculated with integral equation theories based on the Ornstein-Zernike equation and a complementary closure relation. One such closure relation is the hypernetted chain (HNC) approximation, which neglects the so-called bridge function. We present a way to use machine learning to train a deep operator network to predict the bridge function, based on the radial distribution function as input. Bridge functions for the Lennard-Jones fluid are calculated from Monte Carlo simulations in a wide range of densities and temperatures. These results are used to train the deep operator network. This network is employed to improve the HNC closure by the prediction for the bridge function, and the resulting set of equations is solved iteratively. For assessment, we compare the radial distribution function and the pressure, calculated by the viral expression, with Monte Carlo results and standard HNC. We demonstrate that incorporating the neural-networkbased bridge function in the closure relation leads to substantially improved predictions. The universality of our method is demonstrated by comparing results for the Mie fluid and the hard-sphere fluid, calculated with our model trained on the Lennard-Jones fluid, with exact Mie and hardsphere results, showing overall good agreement.

DOI: 10.1103/4962-9rch

I. INTRODUCTION

Understanding the structural and thermodynamic properties of atomic and molecular liquids remains a fundamental task in statistical mechanics. The integral equation theory provides an efficient means of describing these properties, with the Ornstein-Zernike (OZ) equation serving as the link between the total correlation function $h(r)$ and the direct correlation function $c(r)$ [1]. Based on the OZ equation, various successful approaches such as the molecular Ornstein-Zernike (MOZ) equation and the reference interaction site model (RISM) have been developed and used, e.g., to study solvation thermodynamics [2].

To solve the OZ equation, various closure relations have been proposed, among which the hypernetted-chain (HNC) approximation plays a significant role [3]. The HNC approximation incorporates essential correlation effects while maintaining a relatively simple mathematical structure. However, it neglects certain higher-order many-body contributions, which are crucial for accurately modeling dense and strongly interacting fluids in a more general way [4].

Diagrammatic expansions provide a systematic approach for deriving closure relations. By considering the Mayer cluster expansion and its graphical representation, one can identify different classes of diagrams that contribute to the correlation functions. The HNC approximation emerges naturally from a subset of these diagrams, but completely neglects so-called bridge diagrams defined below, which account for additional many-body effects [5]. Only in the limit of summing up the infinite set of all bridge diagrams would the resulting bridge function and thus the integral equation theory become exact.

The bridge diagrams are a well-defined set of cluster diagrams where the connections (bonds) are the Mayer function, $f(r) = e^{-\beta v(r)} - 1$, where $\beta = \frac{1}{k_B T}$ and we assume a pairwise interaction between point particles (such as interaction sites in RISM) given by $\sum_{i<j} v(r_{ij})$. Resummation of subdiagrams can greatly reduce the number of diagrams. For the bridge diagrams this leads to the redefinition of the bonds as the total correlation function $h(r)$. In this way we see that the bridge function is a functional of $h(r)$ or, equivalently, of the radial

[*]Contact author: panholzer@unisoftwareplus.com
[†]Contact author: thomas.wallek@tugraz.at
[‡]Contact author: robert.zillich@jku.at

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

distribution function (RDF) $g(r) = h(r) + 1$. Exact knowledge of the bridge functional $B(r) = B[g; r]$ would allow the exact calculation of $g(r)$ for a given interaction $v(r)$—and from that essentially all thermodynamic properties—by self-consistently solving the exact closure [1], which we call the HNCB equation,

$$g(r) = \exp\left[-\beta v(r) + \gamma(r) + B(r)\right], \tag{1}$$

and the OZ equation,

$$\gamma(r) = \rho \int d^3r' \left(g(r' - r) - 1\right)c(r), \tag{2}$$

where $\gamma(r) = g(r) - 1 - c(r)$ is the indirect correlation function. Hence, in principle the integral equation method provides a framework for calculating thermodynamic properties of fluids exactly. The catch is that the exact bridge functional $B[g; r]$ cannot be calculated because it is an infinite sum of Mayer cluster diagrams. Usually, $B[g; r]$ has been approximated, leading to a host of approximate closures of the OZ equation [6–13]. Unfortunately, these approximations are usually only suitable for a certain type of liquid (Coulomb [14], hard sphere (HS), Lennard-Jones (LJ), etc.). The trivial approximation is to omit B completely, $B(r) = 0$, i.e., the above-mentioned HNC approximation.

Monte Carlo (MC) and molecular dynamics (MD) simulations serve as approaches complementary to integral equation theory by providing numerical benchmarks for correlation functions and thermodynamic properties [15], but they require orders-of-magnitude-higher computational effort than integral equation methods. Molecular simulations allow for direct sampling of particle configurations and can be used to validate theoretical predictions obtained via integral equation theories.

Recently, various attempts have been made to link integral equation theories and molecular simulations with the machine learning (ML) methods from the rapidly evolving field of artificial intelligence (AI). Such methods have been proposed for various goals, such as to improve the results of established RISM closures with ML-based methods, to directly model RDFs and correlation functions, to model bridge functions, or to extract force-field parameters as part of the so-called inverse problem of statistical mechanics [16,17].

ML-based improvement of RISM results. Osaki *et al.* proposed the three-dimensional (3D) RISM AI method, using the hydration free energy based on the 3D RISM method as input feature for a ML approach to predict the binding free energy for protein-ligand pairs [18]. In their pyRISM-CNN method, Fowles and Palmer combined a one-dimensional (1D) convolutional neural network (CNN) trained on RISM correlation functions with a 1D RISM solver to predict the solvation free energy and thermodynamic parameters of small organic and druglike molecules [19].

ML-based prediction of RDFs and correlation functions. Craven *et al.* used the results of MD simulations to train a ML model based on segmented linear regression and multivariate function decomposition for the prediction of RDFs for pure-component LJ fluids [20]. Li *et al.* predicted the RDFs of Ar, NO, and H_2O from a single MD simulation trajectory using a point-cloud-based deep learning strategy [21]. Carvalho and Braga trained a Hopfield neural network (HNN) to solve the OZ equation and to predict RDFs and direct correlation functions for liquid neon, liquid metal, and a glassy solid from experimental static structure factors determined by neutron scattering experiments [22–24]. Later, these authors also applied a physics-informed neural network (PINN) for the same task to describe argon [25]. To further enhance this PINN approach, Chen *et al.* proposed a PINN tailored for fixed-parameter OZ equations and a physics-informed deep operator network (PIDeepOnet) approach for parametrized OZ equations and reported advances in numerical stability and reduced learning effort [26].

ML-based prediction of bridge functions. Here, the starting points for training data are bridge functions calculated from molecular simulation data using the RDF and cavity correlation function (CCF). Such calculation schemes have been investigated for different model systems, such as HS fluids [27,28], LJ fluids [29–31], Stockmayer fluids [32], liquid metals [33,34], and plasma [35]. Bedolla *et al.* used a three-parameter modified Verlet bridge function to describe HS and square-well fluids, where the three parameters were determined by an unconstrained optimization with evolutionary computation algorithms, enforcing the isothermal compressibility to be equal when computed from various routes [36]. Goodall and Lee approximated bridge functions using a ML model to determine pairwise interaction potentials from structural information in terms of correlation functions. This model was used to correct the HNC closure [16,37]. Their feature set comprised the total correlation function, the direct correlation function, the RDF in the form of the fluctuation-dissipation theorem, and the gradient of the indirect correlation function. The training set, including the total correlation function and the static structure factor, was determined by MD simulations employing 13 different interaction potentials. Multilayer perceptron (MLP) neural networks were trained to approximate the bridge functions.

The goal of this work is to approach an exact result using HNCB equation (1) without relying on one of the above-mentioned approximations for the bridge functions $B(r)$. The bridge functions are learned from exact MC simulations for a given interaction $v(r)$ and a range of densities ρ and temperatures T. From the MC simulation we obtain the exact RDF $g(r)$ as well as $\gamma(r)$ from OZ equation (2) which allows us to invert Eq. (1) and obtain the exact $B(r)$:

$$B(r) = \ln g(r) + \beta v(r) - \gamma(r). \tag{3}$$

For small r, $v(r)$ is very repulsive and, accordingly, $g(r)$ is very small, which would make MC sampling for small r prohibitively expensive. Therefore, for small r the CCF,

$$y(r) = g(r)\exp[\beta v(r)], \tag{4}$$

is calculated *directly* (see below) and $B(r)$ can be calculated from

$$B(r) = \ln y(r) - \gamma(r). \tag{5}$$

Thus, the exact RDFs $g(r)$ and the corresponding exact $B(r)$ constitute the learning set for training an artificial neural network (ANN). We refer to our method, which is based on the exact HNCB closure of Eq. (1) coupled with machine-learned bridge functions, as HNCB-AI. Remember that the bridge functions $B(r)$ are actually functionals of $g(r)$, i.e.,

$B(r) = B[g; r]$. The need for such a "nonlocal" dependency of $B(r)$ on the correlations g was already stressed in Ref. [37].

The MC simulations to calculate the RDF $g(r)$, the static structure factor $S(k) = 1 + \rho \int d^3 r e^{-i\mathbf{kr}}[g(r) - 1]$, and the cavity correlation function $y(r)$ for a set of thermodynamic parameters ρ and T are explained in Sec. II. We restrict the simulations to the well-studied test bed of statistical physics, the Lennard-Jones liquid, where we give all values in LJ units. The corresponding reduced quantities are indicated with stars, i.e., $T^* = T k_B/\epsilon$ and $\rho^* = \rho\sigma^3$, where ϵ and σ are the strength and range of the LJ interaction. The output of the MC simulations is fed into the ANN to train it for $B[g; r]$. The choice of ANN, the learning strategy, and how to solve the coupled HNCB equation [Eq. (1)] and OZ equation [Eq. (2)] with the $B[g; r]$ ANN are explained in Sec. III. Results are discussed in Sec. IV, and conclusions are drawn in Sec. V.

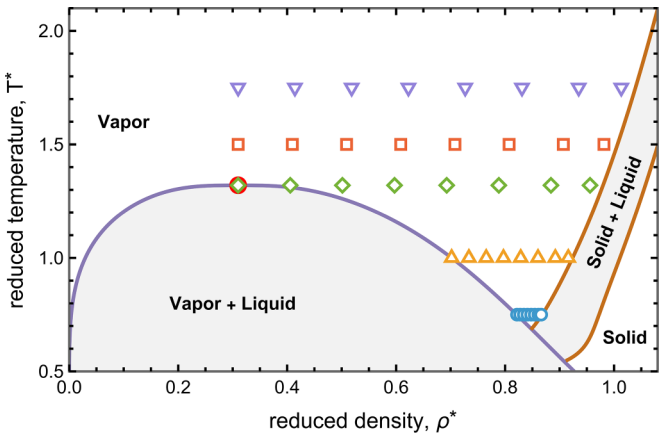


FIG. 1. Points in the phase diagram for which MC calculations were performed. The critical point is marked with a red circle. The phase boundaries, shaded gray, were determined from equations of state [44,45].

II. MOLECULAR SIMULATIONS

Although there is a large amount of published LJ simulation data, it is of limited use for the purposes of this paper: first, because the simulated systems often contain only 500 molecules or fewer, which is significantly below the minimum number of molecules we consider necessary to obtain accurate correlation functions, particularly a static structure factor accurate also for small wave numbers, and second, because the correlation functions are usually only given in figures and not in tabular form, which makes them of little use as training data for machine learning.

For our simulations, we used a MC software suite [38] to conduct canonical ensemble (NVT) simulations of $N = 2048$ LJ particles, starting from a face-centered-cubic lattice and equilibrating for at least 30 000 cycles, where each cycle consisted of N attempted single-particle moves. Samples were then taken over at least 100 000 cycles. We compared the results with simulations of up to 20 000 molecules and found no significant difference in the correlation functions, which is why we considered $N = 2048$ sufficient. The maximum translational distance of attempted single-particle moves was adjusted to achieve a 50% acceptance rate in the Metropolis sampling. Statistical uncertainties were estimated with the method proposed by Flyvbjerg and Petersen [39], using at least 20 remaining samples.

The RDF was sampled using a histogram method [40]:

$$g\left(r + \frac{1}{2}\Delta r\right) = \left\langle \frac{V}{N^2} \frac{1}{V(b)} \sum_{i=1}^{N} \sum_{j \neq i}^{N} h_{\Delta r}(r_{ij}) \right\rangle_{\mathrm{NVT}}. \quad (6)$$

Here, Δr is the bin width, V is the total volume, N is the number of sites, $V(b)$ is the volume of a hollow sphere with inner radius r and width Δr, and $h_{\Delta r}(r_{ij})$ is the characteristic function equal to 1 between r and $r + \Delta r$ and zero otherwise. The histogram approximates the RDF at the center $r + \frac{1}{2}\Delta r$ of the interval $[r, r + \Delta r]$.

Additionally, a state-of-the-art force-based sampling method for the RDF proposed by Rottenberg [41] was used in an attempt to reduce the standard deviation without increasing the simulation length. It was observed that the results from the force-based method had higher standard deviations at radial distances $r > 4\sigma$, which caused inconsistencies during the calculation of the bridge function. Therefore, only simulation results from histogram sampling were used in this work.

The CCF defined in Eq. (4) can be calculated using the Widom particle insertion method [42,43]:

$$y(r) = \langle \exp[-\beta \Psi_{ij}(r)] \rangle_{\mathrm{NVT}} \langle \exp[\beta \Psi_i] \rangle_{\mathrm{NVT}}. \quad (7)$$

Here, Ψ_i is the total interaction energy of a randomly inserted particle i and $\Psi_{ij}(r)$ is Ψ_i without the interaction with a particle j at radial distance r. For small distances $0 < r \leqslant \sigma$, where $g(r)$ is small, we used the direct sampling of the CCF according to Eq. (7), while we calculated $y(r)$ from $g(r)$ using Eq. (4) otherwise.

The static structure factor $S(k)$ can be sampled directly in k space:

$$S(k) = \frac{1}{N} \left\langle \sum_{n=1}^{N} e^{i\vec{k}\cdot\vec{r}_n} \cdot \sum_{m=1}^{N} e^{-i\vec{k}\cdot\vec{r}_m} \right\rangle_{\mathrm{NVT}}. \quad (8)$$

Here, $\vec{r}_n$ is the position of particle n, and the wave vector $\vec{k}$ is chosen such that it is consistent with the periodic boundary conditions $\vec{k} = (n_x, n_y, n_z)2\pi/L$ with integers n_x, n_y, and n_z and the simulation box side length L. We calculated $S(k)$ directly with Eq. (8) for smaller wave vectors, where the periodic boundary restriction is more important, and by Fourier transformation of $g(r) - 1$ for larger k.

We simulated the LJ fluid for 32 combinations of temperature T and density ρ. Figure 1 provides an overview of the simulated state points (T, ρ). The equilibration of the simulations was cross-checked by comparing the resulting bulk properties with those predicted by the LJ equation of state proposed by Thol *et al.*[45], which had been found to be particularly suitable for this purpose when comparing various LJ equations of state in the literature [46]. This validation is shown in Appendix A.

With $g(r)$, $y(r)$, and $S(k)$ thus obtained, we calculated the exact bridge function $B(r)$ by first obtaining the indirect correlation functions $\gamma(r)$ from OZ equation (2), which is

efficiently done by Fourier transformation, and then solved the HNCB equation for $B(r)$ according to Eq. (5).

III. THE HNCB-AI METHOD USING DEEPONETS

A. Architecture

Deep operator networks (DeepONets) are neural net architectures designed to learn mappings between function spaces [47]. In this study, we employ a DeepONet where the trunk network takes as input a scalar variable r, while the branch network processes a function $g(r)$ sampled on a fixed grid with N_G points. The output of the network is a function $B[g;r]$, which depends on both the input variable and the functional input.

The DeepONet consists of two subnetworks: the *trunk network* and the *branch network*. The trunk network encodes the dependency of the output function on the scalar input r. It takes r as input and maps it to a latent representation through a series of fully connected layers with nonlinear activation functions [48]. The branch network processes the function $g(r)$ by sampling it on a fixed grid, $\{r_i\}_{i=1}^{N_G}$, and treating the sampled values $g(r_i)$ as a finite-dimensional input vector. This vector is then passed through multiple layers of a fully connected neural network, extracting relevant features that characterize $g(r)$. For our implementation we used an equidistant grid with $N_G = 200$ points up to a distance of 5σ.

The final output of the DeepONet is obtained by taking the inner product of the feature representations from the trunk and branch networks,

$$B[g;r] = \sum_{i=1}^{N_{\text{net}}} \mathcal{T}_i(r)\mathcal{B}_i(g), \tag{9}$$

where $\mathcal{T}_i(r)$ are the outputs of the trunk network and $\mathcal{B}_i(g)$ are the features extracted by the branch network. $N_{\text{net}} = 100$ was found sufficient for our application.

This DeepONet formulation is particularly useful in learning solution operators for differential equations where the function $g(r)$ represents input parameters, boundary conditions, or source terms. The model efficiently learns complex functional dependencies, making it a powerful tool for scientific computing and engineering applications [49].

In this work we employ a slightly different setting: instead of a differential equation, we use results of a MC simulation to generate the training data.

B. Training

The bases for the training are the bridge functions and the radial distribution functions generated with MC simulation for subsets of the densities and temperatures shown in Fig. 1. The variable r was also randomly sampled in a range from 0.01σ to 9σ with 300 random points uniformly distributed. We added also 200 uniformly distributed points in the region 1σ to 3σ since we found that the features of the bridge function here are sensitive for an accurate result of the RDF.

To ensure a correct limiting behavior in the low-density limit, we added the zero-density limit $g_0(r) = f(r) + 1$ to the training dataset at temperatures $T^* = 0.5, 0.75, 1.0, 1.25, 1.5, 2.0$, and 2.5.

For the training we used an Adam optimizer with a learning rate of 0.001 [50]. As a loss function we used the mean-squared error (squared L_2 norm). We trained for 60 000 epochs and reached a loss below 10^{-4}.

To demonstrate the predictive power of our approach we used ten different training sets. Each training set consists of only 8 randomly selected state points out of the 32 state points obtained with the MC simulations (see Fig. 1). The ten training sets are given in Table II in Appendix B. We denote the trained DeepONet with $B_i[g;r]$, where i numbers the training set. We also trained the DeepONet with all available state points given in Fig. 1 and denote the DeepONet with $B_{\text{LJ}}[g;r]$.

C. Solving the HNCB-AI integral equations

We solve the HNCB-AI equations by generalizing the standard HNC implementation, given, e.g., by Warren *et al.* [51,52], but only using Picard iteration:

(1) Start with a guess for the direct correlation function $c(r)$.

(2) Calculate the indirect correlation function from OZ equation (2), which is efficiently done in Fourier space by $\hat{\gamma}(k) = \frac{\hat{c}(k)}{1-\rho\hat{c}(k)} - \hat{c}(k)$, as described in detail in the literature [43].

(3) Fourier transform the indirect correlation function back to real space, $\gamma(r)$.

(4) Use the HNCB closure relation to obtain an updated direct correlation function $c_{\text{trial}}(r) = \exp[-\beta v(r) + \gamma(r) + B[g;r]] - \gamma(r) - 1$.

(5) To improve convergence, the direct correlation function for the next iteration is obtained by linear mixing of $c_{\text{trial}}(r)$ with the previous $c(r)$. Continue with step 2 until convergence is reached.

The bridge function, needed in step 4, is the prediction with the current RDF calculated as $g(r) = c(r) + \gamma(r) + 1$. For efficiency reasons, the $B[g;r]$ net is only updated every 100th iteration step with the current RDF. To further increase stability of the iteration, we implemented also a linear mixing of the updated bridge function,

$$B_{\text{new}}(r) = B_{\text{old}}(r) + \alpha(B[g;r] - B_{\text{old}}(r)), \tag{10}$$

and with $\alpha = 0.2$ we reached good convergence. Only for three of the reduced training datasets, i.e., 5, 7, and 8, we had a problem with convergence near the critical point. This seems not to be a problem of the solution method; rather for the combined system of HNC and $B[g;r]$ there is no guarantee of a solution if the DeepONet does not behave well enough. In this work we did not investigate which mathematical conditions are required for a unique solution. While important, this topic is outside the scope of this work and is left for future exploration.

IV. RESULTS

We present results for the RDF $g(r)$, the pressure p, and the excess energy U_{ex} obtained with the HNCB-AI method, and compare them with the exact MC results on the one hand and with the simple HNC approximation on the other hand. In Sec. IV A we stay close to the training set consisting of LJ MC simulations by showing only LJ results, and in Sec. IV B

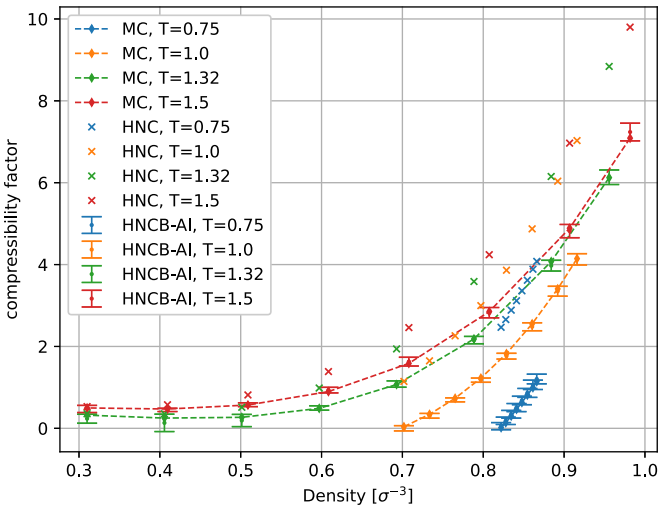


FIG. 2. Comparison between the compressibility factors, p/p_{id}, calculated with the HNCB-AI method using DeepONet approximations for the bridge functionals (symbols with error bars), with the HNC approximation (symbols), and with exact MC simulations (lines). The meaning of the error bars is explained in the text.

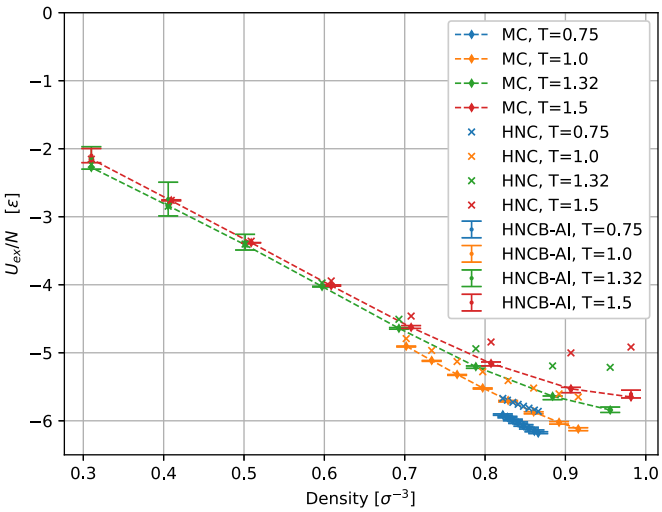


FIG. 3. Same comparison as Fig. 2, but for the excess energies per particle, U_{ex}/N.

we apply the $B[g;r]$ DeepONet for LJ fluids to the Mie fluid and the HS fluid.

A. Lennard-Jones liquid

We incorporate the $B[g;r]$ DeepONet approximation of the bridge function into HNCB equation (1) and solve the coupled equations (1) and (2) self-consistently, as explained in Sec. III C. Since the employed bridge function is only an approximation of the true one, convergence to the correct solution is not guaranteed. Nonetheless, we observe robust convergence across nearly the entire phase diagram, with the exception of regions near the critical point.

As a test of the accuracy of our approach, we applied the HNCB-AI method to the equation of state $p(\rho, T)$. Like essentially all thermodynamic quantities, the pressure p can be calculated from the RDF. We evaluate p using the virial expression

$$p = p_{\mathrm{id}}\left(1 - \frac{2\pi\beta\rho}{3}\int_0^\infty \frac{dv}{dr} g(r) r^3 dr\right), \tag{11}$$

where $p_{\mathrm{id}} = \rho/\beta$ is the ideal gas pressure. Due to its sensitivity to small errors in $g(r)$, Eq. (11) provides a very sensitive test of the predicted RDFs [53].

To assess the validity and robustness of our method, we solve the HNCB-AI equations for all ten models $B_i[g;r]$ and evaluate the pressure. In Fig. 2 we compare the compressibility factor p/p_{id} obtained from HNCB-AI calculations with HNC and MC results. The uncertainty can be quantified by the standard deviation of the ten resulting pressures, indicated by error bars in Fig. 2. The comparison with HNC and MC data clearly demonstrates that incorporating the DeepONet-based bridge function leads to a substantial improvement in the predicted pressure. While for all but the lowest densities ρ the HNC approximation leads to significantly higher pressures than the exact MC results, the HNCB-AI results are almost indistinguishable from the MC results. The only state point where the uncertainty relative to the pressure value becomes large is near the critical point at $T^* = 1.32$ and $\rho^* = 0.31$. Similar results are obtained for the excess internal energy per particle, calculated by

$$U_{\mathrm{ex}}/N = 2\pi\rho\int_0^\infty v(r) g(r) r^2 dr, \tag{12}$$

as shown in Fig. 3. We investigate the quality of the HNCB-AI results further by directly comparing the RDFs between HNCB-AI and MC simulations.

In the top panel of Fig. 4, we show the RDF $g(r)$ for $T^* = 1.32$ and $\rho^* = 0.693$, i.e., at the critical temperature, but far from the critical density (cf. Fig. 1). The solid lines show $g(r)$ (blue) and the associated $B(r)$ (orange) obtained from the MC simulation. The gray dotted lines represent $g(r)$ obtained with our ten models, and are essentially indistinguishable from the MC result, although only in two of the models (2 and 5), the state point was in the training set. Because of the small error between our method and the exact result, we also show the difference between the RDFs in the middle panel of Fig. 4. The relative deviation of the HNCB-AI $g(r)$ from the exact MC $g(r)$ is less than 2%. The DeepONet bridge functions $B(r)$ for the ten models are shown as dash-dotted lines in the bottom panel. They too match the exact MC $B(r)$ almost perfectly, apart from distances r well below σ, which are completely irrelevant for the calculation of $g(r)$ because of the repulsive core of the LJ interaction [see Eq. (1)].

In Fig. 5, we show the same quantities as in Fig. 4, but for $T^* = 1.32$ and $\rho^* = 0.406$, which is much closer to the critical density $\rho^* = 0.310$, and very close to phase separation, where the homogeneous liquid is unstable (cf. Fig. 2). In contrast to the results for the higher density, not all DeepONet models lead to a stable physical solution for $g(r)$: For models which did not contain this state point in the training set, we see a wide variation of $g(r)$ and $B(r)$ between these models. Only the models which do contain this state point in the training set (models 6, 7, and 8) provide reliable results in this case. This large variation of outcomes results in the large relative

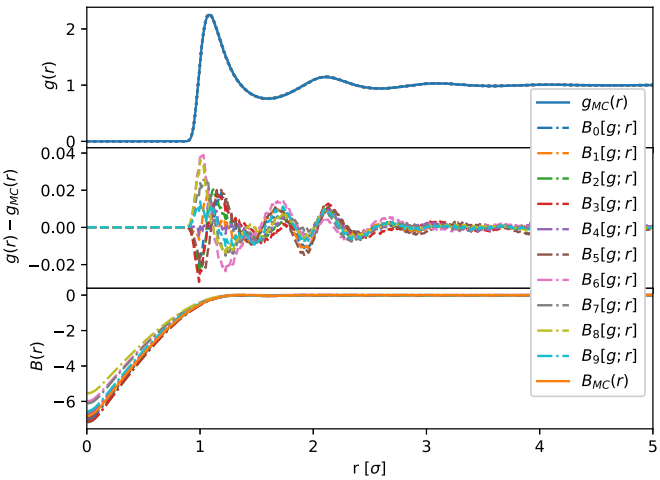


FIG. 4. Comparison of the $g(r)$ results from our HNCB-AI method with DeepONet bridge functions (dotted gray lines) and from MC simulations (solid blue lines) for temperature $T = 1.32\,\epsilon/k_B$ and $\rho = 0.693\sigma^{-3}$, a density far from the critical one. Only in model 0 are the MC data for this density and temperature in the training set. For all ten models, the results for $g(r)$ from the HNCB-AI method are indistinguishable from the MC result in the top panel. Therefore, we also show the difference (dashed lines with different colors). The HNCB-AI results for the bridge function $B(r)$ (dash-dotted lines with different colors) are compared to the MC result in the bottom panel.

uncertainty for the pressure and excess energy visible in Figs. 2 and 3, respectively.

This uncertainty is not entirely unexpected. Critical points, where a system undergoes a second-order phase transition, are notoriously hard to calculate quantitatively. Also, molecular simulations become cumbersome for a LJ fluid close to the critical point due to large density fluctuations over many length scales. Near the critial point, we are close to the spinodal instability where a small change in T or p would break up the homogeneous liquid and lead to phase separation. Small deviations thus have a big effect in this region, and the OZ and

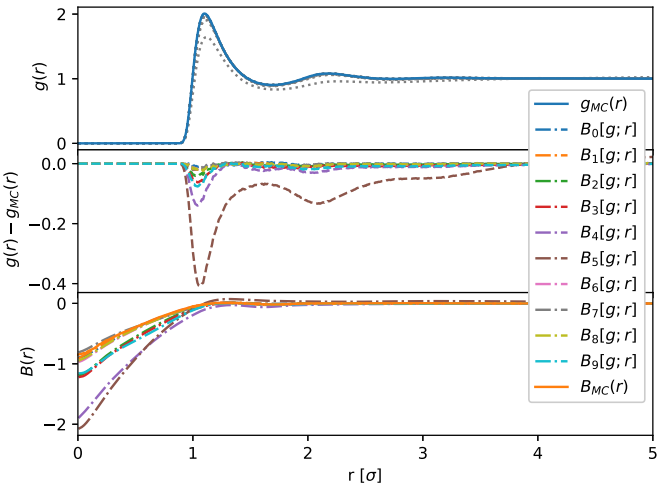


FIG. 5. Same as Fig. 4, but for a density close to the critical point, $\rho = 0.406\sigma^{-3}$. Only in models 6, 7, and 8 are the MC data for this density and temperature in the training set.

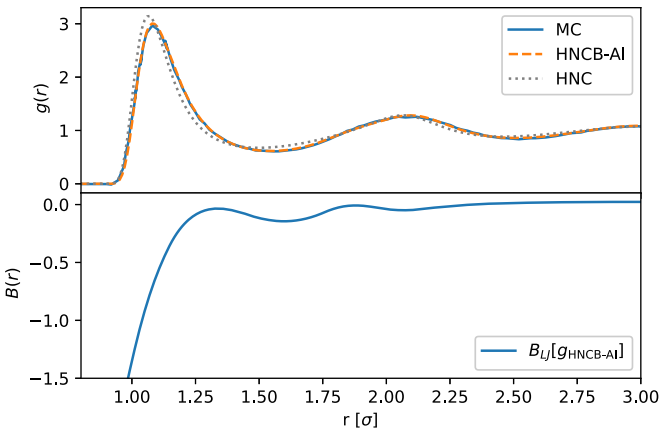


FIG. 6. Top: The HNC and the HNCB-AI RDF for krypton, modeled by a Mie 14-6 potential, at $T = 130$ K and $\rho = 1.6 \times 10^{-2}$ Å^{-3} is compared to MC data [55], where the DeepONet for $B_{\mathrm{LJ}}[g; r]$ was trained only on the LJ model. Bottom: $B_{\mathrm{LJ}}[g; r]$ is plotted.

HNCB equations are harder to solve. Already our MC input for the DeepONet might be afflicted with finite-size effects. A proper treatment would require a finite-size scaling analysis [54], but the detailed study of second-order phase transitions is not within the scope of this paper. Instead, we are more interested in using the HNCB-AI method for many different molecules and solvation phenomena, with a wide range of interactions. Therefore, in the following we test HNCB-AI trained with LJ data on different interactions.

B. Universality of the $B_{\mathrm{LJ}}[g; r]$ DeepONet

In this work, the DeepONet model for the bridge functionals $B_{\mathrm{LJ}}[g; r]$ is based solely on the MC simulation data for the LJ fluid. But since the training data contain no direct information about the interaction potential, one might question the applicability of the functional to other types of interaction.

We first compare our HNCB-AI predictions for the RDF with MC results of Ref. [55], where noble gases were modeled by Mie potentials. We chose krypton for comparison because its RDF differs most from the HNC approximation. The interaction is modeled with a 14-6 Mie potential. The comparison of the MC result with HNCB-AI is shown in Fig. 6. Although the DeepONet knows nothing about the Mie potential, the RDF results from HNCB-AI and MC are nearly indistinguishable.

This motivated us to investigate a system that differs even more strongly from the LJ model, the HS fluid. We solve the OZ and HNCB equations for $g(r)$ for hard spheres with the same interaction range σ as in the LJ potential, using the LJ-trained $B_{\mathrm{LJ}}[g; r]$. Note that, apart from a strong short-range repulsion between particles, the physics of the HS fluid is quite different from the LJ fluid: the HS fluid is not a liquid and as such has no liquid-gas phase transition and particularly no critical point; in fact, temperature is irrelevant for the HS fluid.

In Fig. 7 we show $g(r)$ for the HS fluid obtained with our HNCB-AI method and compare it with exact MC results [27] and the HNC approximation for the three densities $\rho^* = 0.7, 0.8, 0.9$. Our method is in very good agreement with

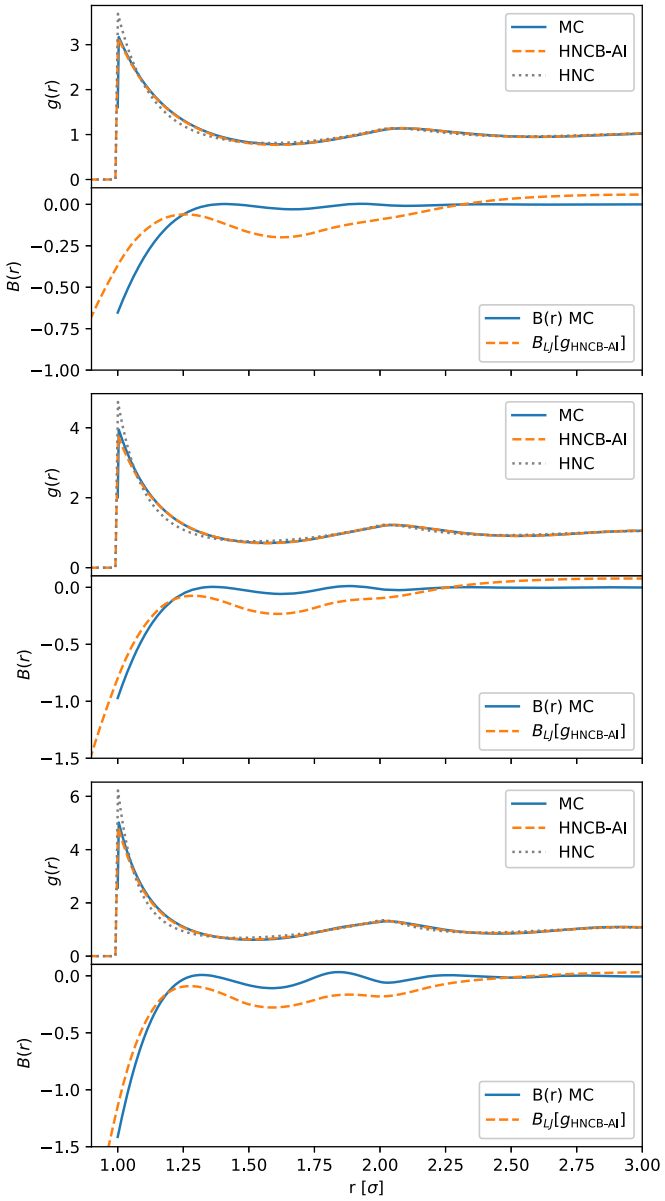


FIG. 7. Hard-sphere results with $B_{LJ}[g;r]$ for densities $\rho^* = 0.7, 0.8, 0.9$ from top to bottom panel. The top half in each panel compares $g(r)$ obtained with MC simulation (blue) with HNCB-AI using the models trained only on the LJ fluid (orange) and with the HNC approximation (grey dashed). The bottom half shows the exact bridge functions $B(r)$ from MC (blue) and from HNCB-AI simulations (orange). MC results are taken from Kolafa *et al.* [27].

the MC result, except close to the HS radius σ, where $g(r)$ exhibits a nonanalytic behavior at $r = \sigma$ and where HNCB-AI slightly underestimates the peak. In contrast, the HNC approximation deviates from the MC result for the whole range of distances r shown in Fig. 7. We further compare the compressibility factors calculated from the contact value of the RDF in Table I (note that for HNC and HNCB-AI the value at $r = \sigma$ was obtained by extrapolation due to the finite r grid). We observe that the addition of the $B_{LJ}[g;r]$ nearly closes the large gap between HNC and the exact value.

All this demonstrates that the HNCB-AI method is very robust against changes to a different interaction potential, which

TABLE I. Comparison of compressibility factors for the hard-sphere system in different approximations. From left to right: HNC, HNCB-AI, MC, and the Carnahan-Starling equation of state [1].

$\rho\sigma^3$	p_{HNC}/p_{id}	$p_{HNCB-AI}/p_{id}$	p_{MC}/p_{id}	p_{CS}/p_{id}
0.7	6.437	5.648	5.716	5.710
0.8	9.006	7.356	7.744	7.750
0.9	12.864	10.229	10.704	10.746

is not in the training set and which even leads to a qualitatively very different $g(r)$ in the case of the HS model. The explanation for this robustness is the many-to-one mapping from rather different RDFs to very similar bridge functions [37]. This apparent universality of $B[g;r]$ has been long noticed and exploited in approximations where the true bridge functions are replaced by known reference bridge functions [4]. This universality is of course not absolute, but the flexibility of the DeepONet ensures that we find an optimal replacement $B_{LJ}[g;r]$ for the true $B_{HS}[g;r]$ without cumbersome adjustment of reference bridge functions.

Indeed, compared to $g(r)$, $B(r)$ is a fairly featureless function and looks rather similar for a LJ liquid and a HS fluid. In Fig. 8 we compare the output of our $B_{LJ}[g;r]$ DeepONet for a HS and a LJ input. We observe that indeed the prediction for the bridge function with the LJ input, $B_{LJ}[g_{LJ};r]$, is very close to the MC result. If we use the HS input, the agreement below 1.25σ is still excellent, but for larger values a substantial difference is visible, which is not surprising as the training data do not contain any information on HS fluids. What is surprising, however, is that the difference in this region has only little influence on the resulting $g(r)$, since part of it is canceled by the indirect correlation function $\gamma(r)$ as we show in Fig. 10 in Appendix C.

V. CONCLUSIONS

This study investigates whether it is possible to train an artificial neural network to predict the bridge functions, which

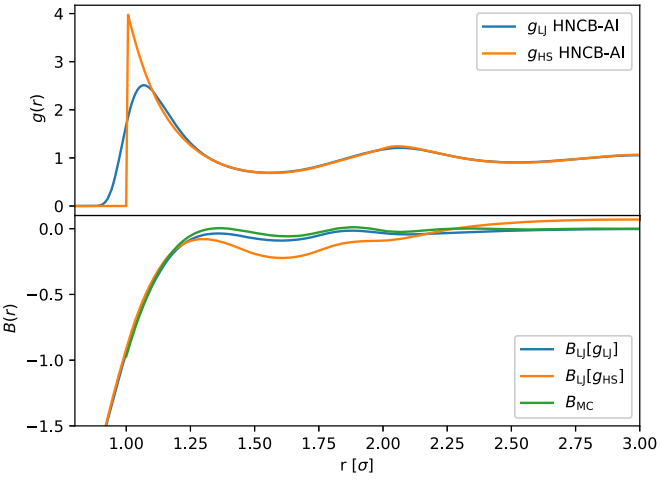


FIG. 8. Top: The RDF for the LJ fluid at $T^* = 1.3$ and $\rho^* = 0.8$ and the HS fluid at the same density are compared, which are input to the $B_{LJ}[g;r]$ net. Bottom: The result is compared to the MC HS bridge function.

is the missing information in the hypernetted-chain (HNC) closure of the integral equation theory to describe a classical fluid exactly. We generate learning data in the form of the RDF $g(r)$ for a single-component Lennard-Jones fluid, with MC simulations which we use as input for a deep operator network, DeepONet. The learning target is the bridge functional $B[g;r]$, which is obtained from $g(r)$ by solving the Ornstein-Zernike (OZ) equation [Eq. (2)] and the HNCB equation [Eq. (3)]. The key idea of our method, HNCB-AI, is that the bridge functional has much less structure than other correlation functions and constitutes only a correction to the HNC approximation where $B[g;r]$ is neglected. Our results clearly demonstrate the capability of this approach, not only within one type of interaction, but also for other interactions *without* additional training.

We simulated the LJ fluid with MC for a range of temperatures and densities (T, ρ), including the critical point, from which we obtain exact RDFs $g(r)$, static structure functions, and cavity correlation functions, required for the calculation of $B[g;r]$. We trained a DeepONet with randomly chosen training sets to get ten models for $B[g;r]$ which we used to self-consistently solve OZ equation (2) and HNCB equation (1) to calculate $g(r)$. The quality of the models was tested by comparing the pressure and the excess energy with the exact MC results for all state points. We find very good agreement, except near the critical point, where we observe a larger variance between the models because the liquid is close to phase separation and small changes thus have a big effect. Also the MC results are less reliable there due to finite-size effects which become very important at the critical point. The result is of course considerably improved if the model contains the critical point in the training data.

For a more detailed analysis, we compare the HNCB-AI and MC results for $g(r)$ and $B[g;r]$, which confirms the excellent agreement except near the critical point where $g(r)$ deviated significantly in one of the models that did not include the critical point. We conjecture that if the training set contains the "problematic" state points, like critical points of a LJ fluid, our HNCB-AI method offers a quantitative description of fluids that rivals much more time-consuming exact MC simulations.

We examined the robustness of the HNCB-AI method by employing the models based only on the LJ fluid to different fluids: krypton described by the 14-6 Mie potential and hard spheres with the same interaction range σ as the LJ particles. Even for very high density, the DeepONet bridge functional learned from the LJ simulations proved to be very universal. The RDFs $g(r)$ obtained with these bridge functionals are almost indistinguishable from the exact $g(r)$, except close to $r = \sigma$ in the HS case, where $g(r)$ has a discontinuity.

Our promising results warrant further investigations into whether HNCB-AI could be a general-purpose closure for integral equation theories, usable, for example, for the RISM method. The robustness of HNCB-AI will be tested with a larger variety of interactions, including Coulomb interactions crucial for liquids of molecules with partial charges. Mixtures of solvents and solutes are of interest for chemical applications, and for that we need to assess the applicability of our scheme to complex molecules and intermolecular interactions with different length scales, unlike the single length scale σ present in the single-component fluids studied in this work. Furthermore, it will be beneficial to incorporate "blind" quality checks, such as thermodynamic self-consistency [56], to gauge the validity of HNCB-AI results without doing expensive MC simulations every time a new molecule is studied that is not contained in the learning data.

ACKNOWLEDGMENTS

Calculations were partially performed using supercomputer resources provided by the Austrian Scientific Computing (ASC) cluster, Project No. 72457. The authors gratefully acknowledge support from NAWI Graz.

DATA AVAILABILITY

The data that support the findings of this article are not publicly available upon publication because it is not technically feasible and/or the cost of preparing, depositing, and hosting the data would be prohibitive within the terms of this research project. The data are available from the authors upon reasonable request.

APPENDIX A: VALIDATION OF MC SIMULATION RESULTS

To validate equilibration of the MC simulation results, we compared these to predictions from the LJ equation of state by Thol *et al.*[45]. Figure 9 shows that the compressibility factor p/p_{id} from MC simulations and from the equation of state are essentially indistinguishable.

APPENDIX B: SUBSETS OF TRAINING DATA

Our training set consists of 32 MC simulations for a range of temperatures T and densities ρ, including the critical point of the liquid-gas phase transition (see Fig. 1). We assess the quality of our HNCB-AI method by using ten different training datasets, each consisting of eight randomly chosen phase

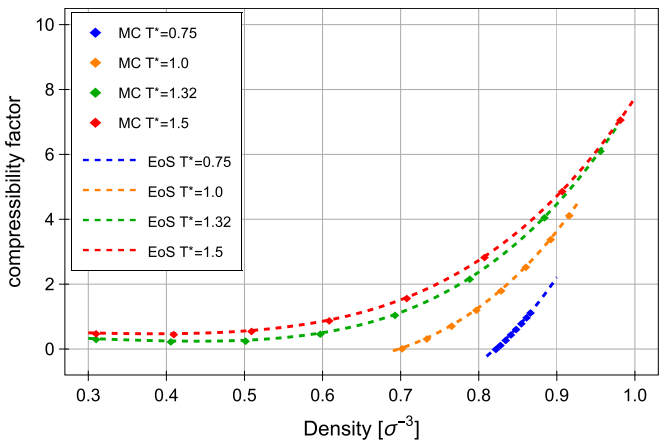


FIG. 9. Comparison between the compressibility factor, p/p_{id}, calculated with the LJ equation of state by Thol *et al.*[45] (lines) and the MC simulation results (symbols).

TABLE II. List of randomly selected MC data for training the ten DeepONet models $B_i[g;r]$.

Temperature	Density	0	1	2	3	4	5	6	7	8	9
0.750	0.822						x			x	
0.750	0.828			x	x	x					
0.750	0.835			x	x	x					
0.750	0.841	x				x		x			
0.750	0.848										x
0.750	0.855						x		x	x	
0.750	0.861	x							x		
0.750	0.866							x	x		
1.000	0.702	x									
1.000	0.734			x	x				x		x
1.000	0.765						x	x			
1.000	0.797										
1.000	0.829					x	x			x	
1.000	0.860		x								
1.000	0.892										
1.000	0.916							x			
1.320	0.310			x	x	x					
1.320	0.406							x	x	x	
1.320	0.501	x	x				x	x	x		
1.320	0.597			x	x	x					
1.320	0.693	x									
1.320	0.789						x				
1.320	0.884							x			
1.320	0.956		x			x					x
1.500	0.310	x					x	x		x	x
1.500	0.409		x								x
1.500	0.509	x	x							x	x
1.500	0.609						x				
1.500	0.708	x		x	x						
1.500	0.807		x						x		
1.500	0.907		x	x	x					x	x
1.500	0.982		x	x	x	x			x	x	x

points (T, ρ), i.e., 25% of the full set of MC simulations, resulting in ten DeepONet models $B_i[g;r]$, where $i = 0, \ldots, 9$ denotes the model. This allows us to collect statistics and provide the error bounds in Figs. 2 and 3 for the compressibility factor and the excess energy, respectively, obtained with HNCB-AI. In Table II we list the randomly chosen training datasets for the ten models.

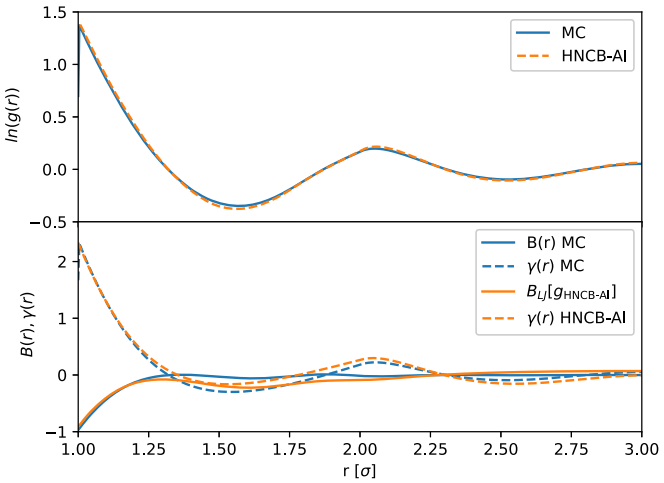


FIG. 10. Top: The logarithm of the RDF is shown for MC and HNCB-AI for $\rho^* = 0.8$, which is the sum of the bridge function and the indirect correlation function $\gamma(r)$, plotted in the bottom panel. It can be seen that the differences in the bridge functions are largely canceled by the corresponding $\gamma(r)$.

APPENDIX C: SENSITIVITY OF THE RDF ON THE BRIDGE FUNCTION

As can be seen in Fig. 7, there are remarkable differences in the hard-sphere bridge function between the MC result [27] and the prediction of the B_{LJ} net, whereas the resulting RDFs show very good agreement. In Fig. 10 we separate the contributions to the RDF $g(r)$ into the bridge function and the indirect correlation $\gamma(r)$ [see Eq. (1)], for $\rho^* = 0.8$. We see that the differences in the bridge functions are largely canceled by opposite differences in $\gamma(r)$, such that the resulting RDF $g(r)$ agrees much better with the exact MC result than the individual contributions.

[1] J.-P. Hansen and I. R. McDonald, *Theory of Simple Liquids with Applications to Soft Matter*, 4th ed. (Elsevier Science, Oxford, 2013).

[2] E. L. Ratkova, D. S. Palmer, and M. V. Fedorov, Solvation thermodynamics of organic molecules by the molecular integral equation theory: Approaching chemical accuracy, Chem. Rev. **115**, 6312 (2015).

[3] T. Morita, Theory of classical fluids: Hyper-netted chain approximation. III: A new integral equation for the pair distribution function, Prog. Theor. Phys. **23**, 829 (1960).

[4] Y. Rosenfeld and N. W. Ashcroft, Theory of simple classical fluids: Universality in the short-range structure, Phys. Rev. A **20**, 1208 (1979).

[5] M. Wertheim, Exact solution of the Percus-Yevick integral equation for hard spheres, Phys. Rev. Lett. **10**, 321 (1963).

[6] J. K. Percus and G. J. Yevick, Analysis of classical statistical mechanics by means of collective coordinates, Phys. Rev. **110**, 1 (1958).

[7] L. Verlet, Integral-equations for classical fluids: II. Hard spheres again, Mol. Phys. **42**, 1291 (1981).

[8] G. A. Martynov and G. N. Sarkisov, Exact equations and the theory of liquids. V, Mol. Phys. **49**, 1495 (1983).

[9] F. J. Rogers and D. A. Young, New, thermodynamically consistent, integral-equation for simple fluids, Phys. Rev. A **30**, 999 (1984).

[10] G. Zerah and J. P. Hansen, Self-consistent integral equations for fluid pair distribution functions: Another attempt, J. Chem. Phys. **84**, 2336 (1986).
[11] D. M. Duh and A. D. J. Haymet, Integral-equation theory for uncharged liquids: The Lennard-Jones fluid and the bridge function, J. Chem. Phys. **103**, 2625 (1995).
[12] A. Kovalenko and F. Hirata, Self-consistent description of a metal-water interface by the Kohn-Sham density functional theory and the three-dimensional reference interaction site model, J. Chem. Phys. **110**, 10095 (1999).
[13] G. A. Martynov, G. N. Sarkisov, and A. G. Vompe, New closure for the Ornstein-Zernike equation, J. Chem. Phys. **110**, 3961 (1999).
[14] M. Baus and J.-P. Hansen, Statistical mechanics of simple Coulomb systems, Phys. Rep. **59**, 1 (1980).
[15] D. Frenkel and B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications*, 3rd ed. (Academic Press, Elsevier, 2023).
[16] R. E. Goodall and A. A. Lee, Data-driven approximations to the bridge function yield improved closures for the Ornstein-Zernike equation, Soft Matter **17**, 5393 (2021).
[17] J. Wu and M. Gu, Perfecting liquid-state theories with machine intelligence, J. Phys. Chem. Lett. **14**, 10545 (2023).
[18] K. Osaki, T. Ekimoto, T. Yamane, and M. Ikeguchi, 3D-RISM-AI: A machine learning approach to predict protein-ligand binding affinity using 3D-RISM, J. Phys. Chem. B **126**, 6148 (2022).
[19] D. J. Fowles and D. S. Palmer, Solvation entropy, enthalpy and free energy prediction using a multi-task deep learning functional in 1D-RISM, Phys. Chem. Chem. Phys. **25**, 6944 (2023).
[20] G. T. Craven, N. Lubbers, K. Barros, and S. Tretiak, Machine learning approaches for structural and thermodynamic properties of a Lennard-Jones fluid, J. Chem. Phys. **153**, 104502 (2020).
[21] C. Li, B. Gilbert, S. Farrell, and P. Zarzycki, Rapid prediction of a liquid structure from a single molecular configuration using deep learning, J. Chem. Inf. Model. **63**, 3742 (2023).
[22] F. S. Carvalho and J. P. Braga, Radial distribution function for liquid gallium from experimental structure factor: A Hopfield neural network approach, J. Mol. Model. **26** (2020).
[23] F. S. Carvalho and J. P. Braga, Indirect solution of Ornstein-Zernike equation using the Hopfield neural network method, Braz. J. Phys. **50**, 489 (2020).
[24] F. S. Carvalho and J. P. Braga, Partial radial distribution functions for a two-component glassy solid, $GeSe_3$, from scattering experimental data using an artificial intelligence framework, J. Mol. Model. **28** (2022).
[25] F. S. Carvalho and J. P. Braga, Physics informed neural networks applied to liquid state theory, J. Mol. Liq. **367**, 120504 (2022).
[26] W. Chen, P. Gao, and P. Stinis, Physics-informed machine learning of the correlation functions in bulk fluids, Phys. Fluids **36**, 017133 (2024).
[27] J. Kolafa, S. Labík, and A. Malijevský, The bridge function of hard spheres by direct inversion of computer simulation data, Mol. Phys. **100**, 2629 (2002).
[28] R. Fantoni and G. Pastore, Computer simulation study of the closure relations in hard sphere fluids, J. Chem. Phys. **120**, 10681 (2004).
[29] T. R. Kunor and S. Taraphder, Molecular dynamics study of the density and temperature dependence of bridge functions in normal and supercritical Lennard-Jones fluids, Phys. Rev. E **72**, 031201 (2005).
[30] T. R. Kunor and S. Taraphder, Bridge functions near the liquid-vapor coexistence curve in binary Lennard-Jones mixtures, Phys. Rev. E **74**, 011201 (2006).
[31] A. R. Saeger, J. K. Johnson, W. G. Chapman, and D. Henderson, Cavity correlation and bridge functions at high density and near the critical point: A test of second-order Percus-Yevick theory, Mol. Phys. **114**, 2516 (2016).
[32] J. Puibasset and L. Belloni, Bridge function for the dipolar fluid from simulation, J. Chem. Phys. **136**, 154503 (2012).
[33] E. Lomba, M. Alvarez, G. Stell, and J. A. Anta, Bridge functions for models of liquid metals, J. Chem. Phys. **97**, 4349 (1992).
[34] S. Kambayashi and J. Chihara, Extraction of the bridge function for simple liquids from a molecular dynamics simulation and its application for correcting the pair distribution function, Phys. Rev. E **50**, 1317 (1994).
[35] P. D. Poll, N. W. Ashcroft, and H. E. DeWitt, One-component plasma bridge function, Phys. Rev. A **37**, 1672 (1988).
[36] E. Bedolla, L. C. Padierna, and R. Castañeda-Priego, Evolutionary optimization of the Verlet closure relation for the hard-sphere and square-well fluids, Phys. Fluids **34**, 077112 (2022).
[37] R. E. A. Goodall and A. A. Lee, Inference of a universal Ornstein–Zernike closure relationship with Machine Learning, in *Conference Book* (Second Workshop on Machine Learning and the Physical Siences (NeurIPS2019), Vancouver, Canada, 2019).
[38] Wolfram Foundation Notebook Archive, https://notebookarchive.org/2022-11-6ger7tx.
[39] H. Flyvbjerg and H. G. Petersen, Error estimates on averages of correlated data, J. Chem. Phys. **91**, 461 (1989).
[40] M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids* (Oxford University Press, Oxford, 2017).
[41] B. Rotenberg, Use the force! Reduced variance estimators for densities, radial distribution functions, and local mobilities in molecular simulations, J. Chem. Phys. **153**, 150902 (2020).
[42] J. Henderson, On the test particle approach to the statistical mechanics of fluids: Hard sphere fluid, Mol. Phys. **48**, 389 (1983).
[43] M. Llano-Restrepo and W. G. Chapman, Bridge function and cavity correlation function for the Lennard-Jones fluid from simulation, J. Chem. Phys. **97**, 2046 (1992).
[44] M. A. Van der Hoef, Free energy of the Lennard-Jones solid, J. Chem. Phys. **113**, 8142 (2000).
[45] M. Thol, G. Rutkai, A. Köster, R. Lustig, R. Span, and J. Vrabec, Equation of state for the Lennard-Jones fluid, J. Phys. Chem. Ref. Data **45**, 023101 (2016).
[46] S. Stephan, J. Staubach, and H. Hasse, Review and comparison of equations of state for the Lennard-Jones fluid, Fluid Phase Equilib. **523**, 112772 (2020).
[47] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators, Nat. Mach. Intell. **3**, 218 (2021).
[48] L. Lu, X. Meng, S. Cai, Z. Mao, S. Goswami, Z. Zhang, and G. E. Karniadakis, A comprehensive and fair comparison of two

neural operators (with practical extensions) based on fair data, Comput. Methods Appl. Mech. Eng. **393**, 114778 (2022).

[49] K. Bhattacharya, B. Hosseini, N. B. Kovachki, and A. M. Stuart, Model reduction and neural networks for parametric PDEs, SMAI J. Comput. Math. **7**, 121 (2021).

[50] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv:1412.6980v9.

[51] P. B. Warren, SunlightHNC, https://github.com/patrickbwarren/SunlightHNC.

[52] P. B. Warren, A. Vlasov, L. Anton, and A. J. Masters, Screening properties of Gaussian electrolyte models, with application to dissipative particle dynamics, J. Chem. Phys. **138**, 204907 (2013).

[53] T. M. Reed and K. E. Gubbins, *Applied Statistical Mechanics: Thermodynamic and Transport Properties of Fluids* (McGraw-Hill, New York, 1973).

[54] K. Binder and D. W. Heermann, *Monte Carlo Simulation in Statistical Physics: An Introduction*, 6th ed. (Springer, Berlin, 2019).

[55] J. R. Mick, M. Soroush Barhaghi, B. Jackman, K. Rushaidat, L. Schwiebert, and J. J. Potoff, Optimized Mie potentials for phase equilibria: Application to noble gases and their mixtures with n-alkanes, J. Chem. Phys. **143**, 114504 (2015).

[56] G. Pellicane, L. L. Lee, and C. Caccamo, Integral-equation theories of fluid phase equilibria in simple fluids, Fluid Phase Equilib. **521**, 112665 (2020).