Analytical classical density functionals from an equation learning network

S.-C. Lin,[1, a)] G. Martius,[2] and M. Oettel[1]

[1)] *Institut für Angewandte Physik, Eberhard Karls Universität Tübingen, 72076 Tübingen, Germany*

[2)] *Max Planck Institute for Intelligent Systems Tübingen, 72076 Tübingen, Germany*

We explore the feasibility of using machine learning methods to obtain an analytic form of the classical free energy functional for two model fluids, hard rods and Lennard–Jones, in one dimension. The Equation Learning Network proposed in Ref. 1 is suitably modified to construct free energy densities which are functions of a set of weighted densities and which are built from a small number of basis functions with flexible combination rules. This setup considerably enlarges the functional space used in the machine learning optimization as compared to previous work[2] where the functional is limited to a simple polynomial form. As a result, we find a good approximation for the exact hard rod functional and its direct correlation function. For the Lennard–Jones fluid, we let the network learn (i) the full excess free energy functional and (ii) the excess free energy functional related to interparticle attractions. Both functionals show a good agreement with simulated density profiles for thermodynamic parameters inside and outside the training region.

[a)] Electronic mail: shang-chun.lin@uni-tuebingen.de

I. INTRODUCTION

Density functional theory (DFT) may be viewed as a great reductionist scheme for classical and quantum many–body systems in equilibrium. The one–to–one correspondence between the one–body density profile of particles and the one–body external potential acting on these particles entails that a unique (free) energy functional of the one–body density contains all of the homogeneous and inhomogeneous equilibrium structure in a given system, and no explicit knowledge of higher–order correlations (i.e. through the phase space distribution of classical particles or the full many–body quantum wavefunction) is needed.

In general, the analytical form of the (free) energy functional is unknown, except for a handful of particular model systems (mostly in one dimension [1D]). In recent years, some effort has gone into approximating ("learning") functionals by machine learning (ML) techniques. In quantum DFT, e.g., interpolating functionals generated by kernel ridge regression have been tested for model 1D systems[3,4] and also have been extended to 3D systems[5]. Numerically interpolated functionals do not contain sufficient information about functional gradients, therefore both the energy–density map and the external potential–density maps had to be learned by interpolation[4]. For the 1D Hubbard model, a convolutional network functional has been learned whose numerical functional derivative appears to be more robust[6]. However, these approaches hide the energy functional inside an "ML black box" which does not permit much insight from a theory perspective. For the classical case, a 1D LJ like fluid was studied with a convolutional network[2] , utilizing an established approach from liquid state theory of splitting the excess free energy functional into a "repulsion" part and an "attraction" part $\mathscr{F}^{\text{ex}}_{\text{att}}$[2]. The convolutional network naturally led to an approximation of $\mathscr{F}^{\text{ex}}_{\text{att}}$ in terms of weighted densities n_i, which are the essential building blocks in modern classical DFT; however, the free energy density $\mathfrak{f}^{\text{ex}}_{\text{att}}(n_i)$ as a function of n_i had to be prescribed as simple polynomials. An interpretable results obtained in[2] was the accurate splitting of the interaction potential in the Weeks–Chandler–Andersen (WCA) spirit[7].

In this context, the question naturally arises whether ML techniques can be used to learn analytic forms of (free) energy functionals instead of "black boxes" or presumed forms. This question is important also in a more general context: can ML algorithms contribute to theory building in physics? In the ML community, efforts in that direction have utilized genetic algorithms to search a space of simple basis function with multiplication and addition rules[8]. More recent work proposes an equation learning network employing gradient-based optimization with simple basis functions and division besides multiplication/addition as combination rules[1,9]. An empirical principle for the "right" formula (choose the simplest one that still predicts well, i.e. Occam's razor) can be built into the cost function. This principle was also successful in the history of physics in finding analytical models with high predictive power even outside the training/observed regime. For the DFT problem, the extrapolation power to other external potentials is an important aspect, as well as the analytic differentiability of the free energy functional since structural information about the fluid (pair correlations) is obtained via the direct correlation function (two functional derivatives of the excess free energy functional). These aspects are explored below for the model cases of a hard rod (HR) and a Lennard–Jones (LJ) fluid in 1D.

II. CLASSICAL DFT

In classical DFT[10–13], the grand potential functional of 1D system is

$$\Omega[\rho(x)] = \mathscr{F}^{\mathrm{id}}[\rho(x)] + \mathscr{F}^{\mathrm{ex}}[\rho(x)] + \int dx (V^{\mathrm{ext}}(x) - \mu)\rho(x), \tag{1}$$

where $\rho(x)$ is the particle density distribution, $\mathscr{F}^{\mathrm{id}}$ is the free energy functional of the ideal gas, $\mathscr{F}^{\mathrm{ex}}$ is the excess free energy functional (unique for a given pair potential between particles), μ is the chemical potential and V^{ext} is an external potential. The exact form of $\mathscr{F}^{\mathrm{id}}$ is:

$$\beta\mathscr{F}^{\mathrm{id}} = \int dx\, \rho(x) \left[\ln(\rho(x)\lambda) - 1\right] \tag{2}$$

with $\beta = 1/(k_{\mathrm{B}}T)$, T the temperature, k_{B} Boltzmann's constant, and λ the thermal wavelength. In the following we set $\beta = \lambda = 1$.

In equilibrium, the corresponding density profile ρ^{eq} minimizes Ω for a given μ. Thus, with $\frac{\delta\Omega}{\delta\rho} = 0$ and Eq. (2), we obtain

$$\rho^{\mathrm{eq}} = \exp\left(\mu - \left.\frac{\delta\mathscr{F}^{\mathrm{ex}}}{\delta\rho}\right|_{\rho=\rho^{\mathrm{eq}}} - V^{\mathrm{ext}}\right). \tag{3}$$

All DFT solutions for test density distributions in this work are obtained by iteratively solving Eq. (3) using the Picard method with mixing.

In this paper, we investigate the HR pair potential:

$$U_{\mathrm{HR}}(x) = \begin{cases} \infty & \text{if} \quad x < \sigma \\ 0 & \text{otherwise} \end{cases}$$

as well as the LJ(–like) potential:

$$U_{\mathrm{LJ}}(x)=\begin{cases}\infty & \text{if} \quad x<\sigma \\ 4\,\epsilon\left[\left(\frac{\sigma}{x}\right)^{12}-\left(\frac{\sigma}{x}\right)^{6}\right] & \text{if} \quad \sigma<x<16\sigma \\ 0 & \text{otherwise}\end{cases}$$

with x the distance between particle centers, σ the diameter of the particles and ϵ the strength of interaction. In the following we set $\sigma=1$.

III. MACHINE LEARNING

A. Model

We define a machine learned excess free energy functional $\mathscr{F}^{\mathrm{ex,ML}}$ and the resulting ML output density ρ^{ML} by

$$\rho^{\mathrm{ML}}(x)=\exp\left(\mu^{\mathrm{ML}}-\frac{\delta\mathscr{F}^{\mathrm{ex,ML}}}{\delta\rho}\bigg|_{\rho=\rho^{\mathrm{eq}}}-V^{\mathrm{ext}}\right). \tag{4}$$

This is the equivalent of a "generative step" of a learned distribution ρ^{ML} from an input distribution ρ^{eq} in an ML network (via weighted densities n_i, see below). Here, μ^{ML} is included in the training process to facilitate convergence. In the end, $\mu^{\mathrm{ML}}\to\mu$, see also Ref. 2 and SI for details. The test density profiles shown in Figs. 2–4 are obtained by initializing with a constant value and then iteratively minimizing the learned functional with the (physical) μ.

The network we propose, Functional Equation Learner (FEQL), is a L–layered feed–forward network with computational units specifically designed for constructing the free energy functional (see Fig. 1). The first layer consists of convolution kernels which compute the weighted densities n_i with the convolution kernel ω_i ($i=1\ldots n_{\mathrm{w}}$) by

$$n_i(x)=\rho\otimes\omega_i=\int dx'\,\rho(x')\,\omega_i(x-x')\,, \tag{5}$$

and some of the weighted densities are multiplied by ϵ in the case of the LJ fluid. The maximum allowed ranges of the ω_i are $[-4\sigma,4\sigma]$ (HR) and $[-8\sigma,8\sigma]$ (LJ). Using weighted densities instead of the particle density is inspired by the exact HR functional[14] and fundamental measure theory[13,15]. The layer 2 is a linear, all–to–all mapping of the vector (of functions) $n=\{n_i(x)\}$ to the vector

$$z^{(l=1)}=W^{(1)}n \tag{6}$$

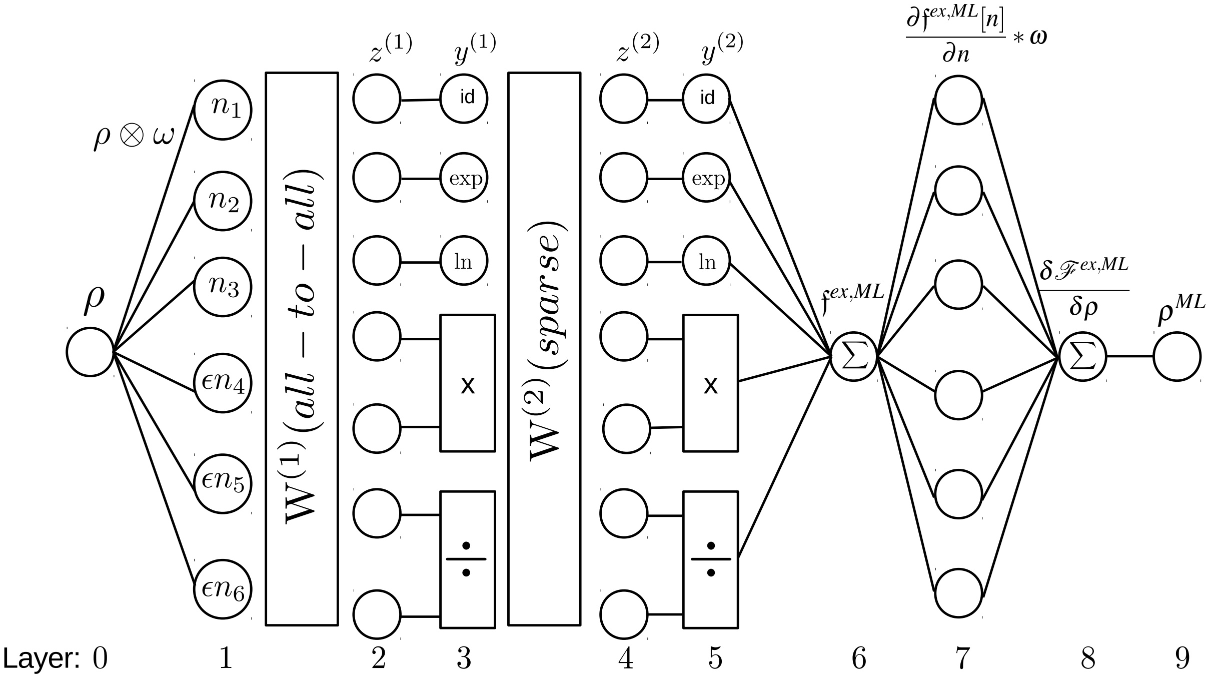


FIG. 1: Network architecture of the proposed FEQL for 10 layers ($L = 9$) and one neuron per type ($u = 3$, $v = 2$) and 6 convolution (weighting) kernels ($n_\mathrm{w} = 6$). ϵ is the coupling strength (equivalent to inverse temperature) in the LJ potential.

at level $l = 1$. The layers $3 \ldots L-4$ are a sequence of nonlinear and linear transformations. The non-linear transformation at level l contains u unary units f_I and v binary units g_J and maps $z^{(l)}(x)$ ($u+2v$–dimensional) to the layer output $y^{(l)}$ ($u+v$–dimensional) as:

$$y^{(l)} := \left(f_1\left(z_1^{(l)}\right), f_2\left(z_2^{(l)}\right) \ldots f_u\left(z_u^{(l)}\right), \right.$$
$$\left. \left(g_1\left(z_{u+1}^{(l)}, z_{u+2}^{(l)}\right) \ldots g_v\left(z_{u+2v-1}^{(l)}, z_{u+2v}^{(l)}\right)\right)\right). \tag{7}$$

The unary units, $f_1, \ldots, f_u$ receive the respective component, $z_1, \ldots, z_u$ as inputs, and each unit is one of the following base functions indexed by $I \in 0, 1, 2$:

$$f_I(z_i) := \begin{cases} z_i & \text{if} \quad I = 0 \\ \exp(z_i) - 1 & \text{if} \quad I = 1 \\ \ln(z_i + 1) & \text{if} \quad I = 2 \end{cases}$$

The binary units, $g_1,\dots,g_v$ receive the remaining component, $z_{u+1},\dots,z_{u+2v}$, as input in pairs of two, and each unit may be multiplication or division indexed by $J \in 0,1$:

$$g_J(z_i,z_{i+1}) := \begin{cases} z_i \times z_{i+1} & \text{if} \quad J=0 \\ z_i \div (z_{i+1}+1) & \text{if} \quad J=1 \end{cases}$$

Note that $f_I(0) = g_J(0,z) = 0$. One may worry about divergences in division and logarithm when $z \to -1$. In the beginning of the training procedure, all parameters and convolution kernels are initialized by small numbers so thus z is close to zero and there are no problems. If z is too close to -1, the loss will change drastically; thus the network will intrinsically handle this issue. As mentioned in Ref. 9, one could use modified division and logarithm functions and add extra penalties. However, it turns out not to be required here.

The linear transformation from level l to $l+1$ maps the $(u+v)$–dimensional input $y^{(l)}$ to the $(u+2v)$–dimensional intermediate representation $z^{(l+1)}$ given by

$$z^{(l+1)} = W^{(l+1)} y^{(l)}. \tag{8}$$

Thus, the n_w convolution kernels $\omega(x)$ in the first layer and the matrices $W^{(l)}$ are free parameters that are learned during training.

The machine-learned free energy density $\mathfrak{f}^{\mathrm{ex,ML}}$ is a summation of the output of layer $L-4$, the functional derivative $\frac{\delta \mathscr{F}^{\mathrm{ex,ML}}}{\delta \rho} = \sum_i \frac{\partial \mathfrak{f}^{\mathrm{ex,ML}}}{\partial n_i} * \omega_i$ (with $\mathscr{F}^{\mathrm{ex,ML}} = \int dx \mathfrak{f}^{\mathrm{ex,ML}}(n)$ and $*$ denoting cross–correlation) is used in the final, generative step (Eq. (4)). More details about constructing FEQL can be found in the SI.

B. Network training

To obtain training data for ρ^{eq}, grand canonical simulation are used in the case of LJ fluid; for the HR fluid, Eq. (3) is directly solved, since the exact functional is known.

FEQL is fully differentiable in its free parameters $\theta = [W,\omega]$ and can thus be trained using back–propagation. We adopt the following loss function

$$\begin{aligned} L = \frac{1}{N}\sum_{i=1}^{N}\left(\alpha_1 \int |\rho_i^{\mathrm{eq}} - \rho_i^{\mathrm{ML}}|\, dx + \alpha_2 |\mu_i^{\mathrm{eq}} - \mu_i^{\mathrm{ML}}|\right) \\ + \lambda_1 \sum_i \int dx\, |\omega_i| + \lambda_2 \sum_{l,\beta\gamma} |W_{\beta\gamma}^{(l)}|, \end{aligned} \tag{9}$$

with $\alpha_1 = 0.9$ and $\alpha_2 = 0.1$. These values have been determined empirically and the exact choice is not critical. For training we choose Adam[16] with mini-batches:

$$\theta_{t+1} = \theta_t + \mathrm{Adam}\left(\frac{\partial L(D(t))}{\partial \theta}, \alpha\right) \tag{10}$$

with α the stepsize parameter (learning rate) and $D(t)$ the data in the current mini-batch. The choice of Adam is not critical and standard stochastic gradient descent also works.

Following Sahoo *et al.*[9], we adopt a three–step training procedure. At the beginning, we use no regularization ($\lambda_1 = \lambda_2 = 0$), such that parameters can vary freely and reach reasonable starting points. In step 2, we switch on the regularization by setting λ_1 and λ_2 to positive finite values to sparsify the network for obtaining a simpler functional. In step 3, we clamp small parameters with $|W^{(l)}_{\beta\gamma}| < w_{\mathrm{th}}$ to zero. In this way we keep the sparsity introduced by the lasso[17] training in step 2 but make sure unbiased parameter values are attained. In this paper we choose $\alpha = 10^{-2}$ or 10^{-3}, $\lambda_1 = 10^{-7}$ and $w_{\mathrm{th}} = 0.05$.

IV. RESULT

A. Hard rods

The exact equation of state (eos) for the hard rod (HR) fluid is given by the pressure $P(\rho) = \frac{\rho}{1-\rho}$ and the analytic form of $\mathscr{F}^{\mathrm{HR}}$ (Percus functional) is one of the few exactly known ones[14,18].

The parameter of $\mathscr{F}^{\mathrm{ex,ML}}$ are trained using 1024 density profiles in a hard wall slit of width 32 σ with 3 additional Gaussian potentials of random strength/width and location inside the slit and with a range of training reservoir densities $\rho_0 = 0.2 \ldots 0.55$. We choose $n_{\mathrm{w}} = 3$ and (1,1,1,3,1) nodes for (identity, exponential, logarithm, multiplication and division) with $L = 10$ layers (see Fig. 1) and $\lambda_2 = 8 \cdot 10^{-5}$ in Eq. (9) (results for different λ_2 and arguments for an optimal choice are shown in the SI). $\mathscr{F}^{\mathrm{ex,ML}}$ is not of the form of the Percus functional, since the convolution kernels of the latter are Dirac delta and Heaviside step functions, which are are hard to be captured by our network.

In Fig.2, we show the eos, a density profile inside the thermodynamic training region but not in the training data ($\rho_0 = 0.49$) and a density profile outside the training region ($\rho_0 = 0.80$). The FEQL recovers the almost exact result inside the training region and also performs quite well even outside the training region. The ML density profiles are initialized by $\rho = 0.5$ and then iteratively solved using Eq. (3) with $\mathscr{F}^{\mathrm{ex}} = \mathscr{F}^{\mathrm{ex,ML}}$.

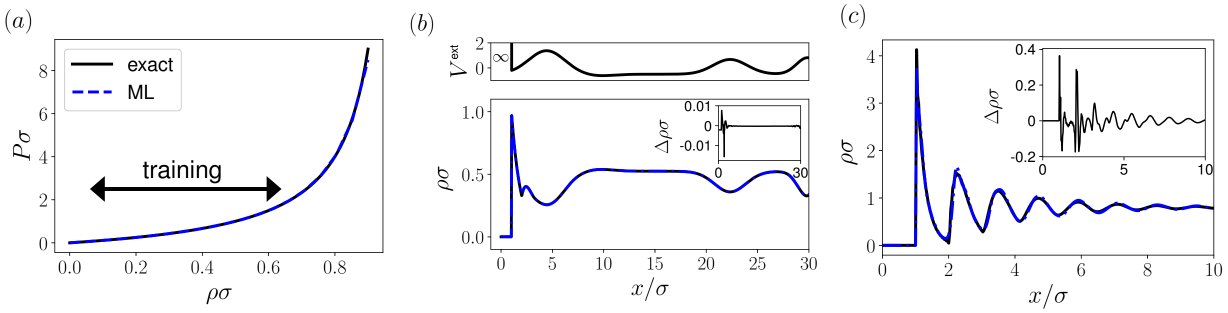


FIG. 2: FEQL results for hard rods. Dark solid lines are exact solutions from $\mathscr{F}^{\text{HR}}$ and blue dashed lines are ML results. (a) eos, $P(\rho)$. (b) density profile for $\rho_0 = 0.49$ inside the training region but V^{ext} not in the training data. (c) density profile at hard wall for $\rho_0 = 0.80$ outside the training region. Insets in (b) and (c) show $\Delta\rho = \rho^{\text{exact}} - \rho^{\text{ML}}$.

The virial expansion

$$P^{\text{ML}}(\rho) \simeq \rho + 1.03\rho^2 + 0.71\rho^3 + O\left(\rho^4\right) \tag{11}$$

of the ML eos shows moderate deviation compared to the exact one ($\frac{\rho}{1-\rho} = \rho + \rho^2 + \rho^3 \ldots$). One sees that inaccuracies in these coefficients do not necessarily mean a poor eos. Despite inaccurate coefficients, the higher order terms in the learned eos combine appropriately to give a good representation of the exact eos. This is understandable since no explicit information about virial coefficients is incorporated into the cost function, thus the learning procedure has little incentive to find the correct coefficients.

B. Lennard–Jones

Here, 1115 training distributions are generated with random μ and ϵ in the range of $0.5 \ldots 1.5$ and $\ln 0.5 \ldots \ln 2$, respectively, with V^{ext} prescribed as in the hard rod case. The training data are obtained by grand canonical Monte Carlo (GCMC) simulation.

1. *Splitting between repulsions and attractions*

Following liquid state theory[2], we split into a contribution from repulsions and one from attractions as follows:

$$\mathscr{F}^{\text{ex,ML}}([\rho];\epsilon) = \mathscr{F}^{\text{HR}}([\rho]) + \epsilon\, \mathscr{F}^{\text{ex,ML}}_{\text{att}}([\rho];\epsilon), \tag{12}$$

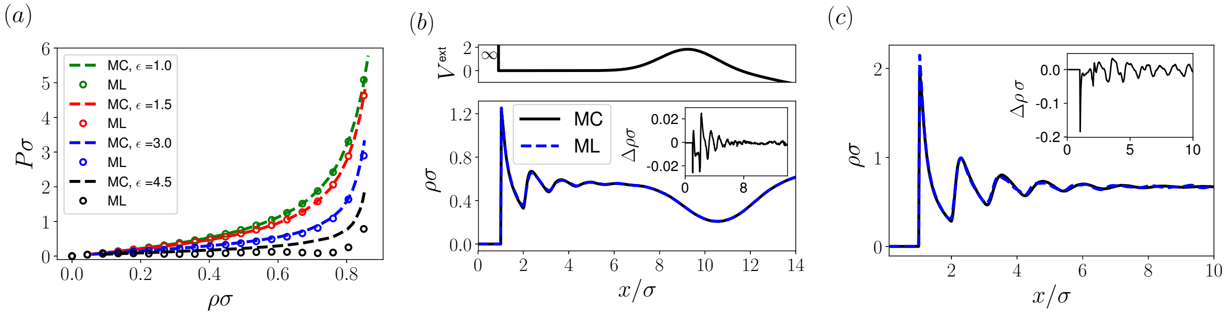


FIG. 3: FEQL results for LJ fluid with functional splitting. (a) eos $P(\rho)$. (b) density profile for $\epsilon = 1.30$, $\mu = \ln(1.27)$ inside the training region but V^{ext} not in the training data. (c) density profile at a hard wall for $\epsilon = 1.7$, $\mu = \ln(1.7)$, outside the training region. Dark solid lines are simulation profiles and blue dashed lines are ML results. Insets in (b) and (c) show $\Delta\rho = \rho^{\mathrm{MC}} - \rho^{\mathrm{ML}}$.

where the factor ϵ in front of the $\mathscr{F}_{\mathrm{att}}^{\mathrm{ex,ML}}$ makes sure $\mathscr{F}^{\mathrm{ex,ML}}(\epsilon \to 0) = \mathscr{F}^{\mathrm{HR}}$, and $\mathscr{F}_{\mathrm{att}}^{\mathrm{ex,ML}}$ is to be learned by the network. In Fig. 1, the output from the layer 6 is $\mathscr{F}_{\mathrm{att}}^{\mathrm{ex,ML}}$; we multiply the output from the layer 8 by ϵ, add contribution from $\mathscr{F}^{\mathrm{HR}}$, and then feed it to the layer 9. In the first layer, we choose $n_{\mathrm{w}} = 4$, 1 kernel multiplied by ϵ and another 3 without this factor (see Fig. 1), and (1,1,1,2,1) nodes for (identity, exponential, logarithm, multiplication and division). The training parameter $\lambda_2 = 5 \cdot 10^{-5}$ in Eq. (9). Results are shown in Fig. 3. The findings are similar to the HR case with a very good match to simulation data for the eos and test distributions inside and outside the thermodynamic training region. For a 1D system with hard–core repulsive and finite range attractive pair potential, the pressure must be monotonically increasing for arbitrary low temperature (high ϵ), and thus resulting no gas–liquid transition[19]. The corresponding ML pressure shows no van der Waals (vdW) loop for attractions strengths up to $\epsilon = 4.1$, this is a qualitative step forward as compared to Ref. 2.

2. *No splitting*

As a further test of the capability of FEQL, we forego the splitting of the functional such that $\mathscr{F}^{\mathrm{ex}} = \mathscr{F}^{\mathrm{ex,ML}}$. In the first layer, we choose $n_{\mathrm{w}} = 6$, 3 kernels multiplied with ϵ and another 3 without this factor (see Fig. 1), and (1,1,1,3,1) nodes for (identity, exponential, logarithm, multiplication and division). The training parameter $\lambda_2 = 5 \cdot 10^{-5}$ in Eq. (9). For the training data we also include density profiles from the HR case. In Fig. 4, we show the results. Test distributions match well

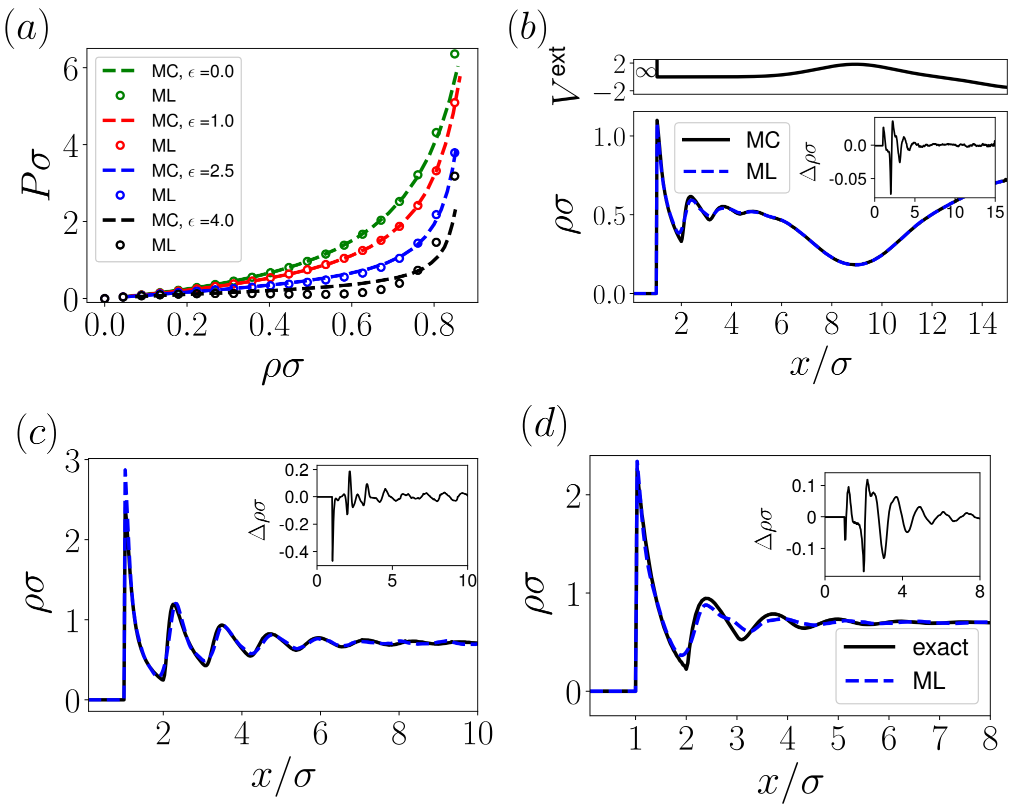


FIG. 4: FEQL results for LJ fluid (no splitting). (a) eos $P(\rho)$. (b) density profile for ϵ= 1.25, $\mu = \ln(1.15)$ inside the training region but V^{ext} not in the training data. (c,d) density profile at a hard wall for ϵ= 1.9, $\mu = \ln(1.9)$ (c) and ϵ= 0, $\rho_0 = 0.7$ (HR limit, (d)). Dark solid lines are simulation profiles and blue dashed lines are ML results. Insets in (b) and (c) show $\Delta\rho = \rho^{\mathrm{MC}} - \rho^{\mathrm{ML}}$ and in (d) $\Delta\rho = \rho^{\mathrm{exact}} - \rho^{\mathrm{ML}}$.

to simulation data both in the HR limit and the regime of higher attractions. The eos shows an unphysical vdW loop for attractions strengths ϵ> 3.7, much higher than the upper limit of the training data.

C. Direct correlation function

The direct correlation function (dcf) is a central object in DFT which through iterations yields the pair corelation function (Ornstein–Zernike relation, see also Chap. 3 in Ref. 10). It is given by the second functional derivative of $\mathscr{F}^{\mathrm{ex}}$:

$$C^{(2)}(x_1, x_2; \rho_0) = -\frac{\beta\delta^2\mathscr{F}^{\mathrm{ex}}}{\delta\rho(x_1)\delta\rho(x_2)}, \tag{13}$$

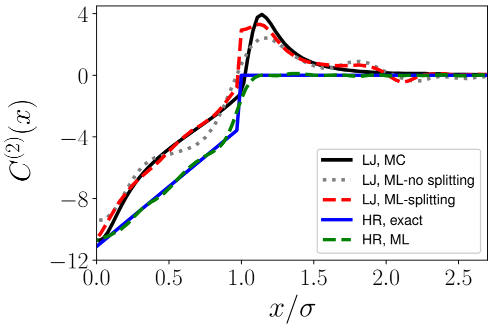


FIG. 5: $C^{(2)}(x,\rho_0)$ with $\rho_0 = 0.7$ (HR) and 0.703 (LJ, ϵ= 1.85 and $\mu = \ln(1.85)$).

and it depends only on $x = |x_1 - x_2|$ in the case of a homogeneous fluid with density ρ_0.

As the network is only trained on the level of the first functional derivative (see Eq. 4), it is a challenge for FEQL to capture the dcf. In Fig. 5, we show exemplary dcf's at moderate to high density for the exact HR functional, LJ from simulation and the corresponding ML results. The direct correlations inside the hard core are captured very well by ML in the HR and LJ cases. Outside the hard core, in the HR case, the $C^{(2)}$ from the ML correctly shows insignificant correlation. In the case of LJ, the contribution to $C^{(2)}$ from attraction is semi–quantitatively correct, with a better result in the splitting case.

V. CONCLUSION

The adaptation of EQL[9] to the classical DFT problem of finding $\mathscr{F}^{\mathrm{ex}}$ has shown satisfactory results for the exemplary case of the 1D HR and LJ fluid. The new network FEQL is very flexible and goes significantly beyond the polynomial *ansatz* used in Ref. 2. The analytic form allows for more easily transferable output and further calculations to obtain, e.g., direct correlation functions. An application to more realistic systems in 3D and perhaps also complex fluids such as water appears to be promising[20–22]. From the results of this work we conclude that the incorporation of results from liquid state theory (separation of repulsion and attraction) is not essential here; however, it increases the reliability and trainability of the ML functional. Future work should include information on virial or high density expansions as well as correlation functions (via test particles) and should develop more quantitative measures for extrapolative capabilities of ML functionals.

SUPPLEMENTARY INFORMATION

See supplementary information for more discussion about the training procedure, FEQL, μ^{ML}, comparison of Ref. 2, explicit $\mathscr{F}^{\mathrm{ex,ML}}$, and convolution kernels $\omega(x)$,

ACKNOWLEDGMENTS

The authors acknowledge support by the High Performance and Cloud Computing Group at the Zentrum für Datenverarbeitung of the University of Tübingen, the state of Baden-Württemberg through bwHPC and the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) through grant no INST 37/935-1 FUGG. Further, We acknowledge the financial support from DFG through the fund ZUK 63 and grant OE 285/5-1 .

REFERENCES

[1] G. Martius and C. H. Lampert, arXiv:1610.02995 (2016).

[2] S.-C. Lin and M. Oettel, SciPost Phys. **6**, 025 (2019).

[3] J. C. Snyder, M. Rupp, K. Hansen, K.-R. Müller, and K. Burke, Phys. Rev. Lett. **108**, 253002 (2012).

[4] L. Li, J. C. Snyder, I. M. Pelaschier, J. Huang, U.-N. Niranjan, P. Duncan, M. Rupp, K.-R. Müller, and K. Burke, Int. J. Quantum Chem. **116**, 819 (2016).

[5] F. Brockherde, L. Vogt, L. Li, M. E. Tuckerman, K. Burke, and K.-R. Müller, Nat. Commun. **8**, 872 (2017).

[6] J. Nelson, R. Tiwari, and S. Sanvito, Phys. Rev. B **99**, 075132 (2019).

[7] M. Bishop, Am. J. Phys. **52**, 158 (1984).

[8] M. Schmidt and H. Lipson, Science **324**, 81 (2009).

[9] S. S. Sahoo, C. H. Lampert, and G. Martius, in *Proceedings of the 35th International Conference on Machine Learning*, Vol. 80 (PMLR, 2018) pp. 4439–4447.

[10] J.-P. Hansen and I. R. McDonald, *Theory of simple liquids: with applications to soft matter* (Academic Press, 2013).

[11] R. Evans, Advances in Physics **28**, 143 (1979).

[12] D. Henderson, *Fundamentals of inhomogeneous fluids* (CRC Press, 1992).

[13] R. Roth, J. Phys. Condens. Matter **22**, 063102 (2010).

[14] J. K. Percus, J. Stat. Phys. **15**, 505 (1976).

[15] Y. Rosenfeld, Phys. Rev. Lett. **63**, 980 (1989).

[16] D. P. Kingma and J. Ba, arXiv:1412.6980 (2014).

[17] R. Tibshirani, J. R. Stat. Soc. Series B Stat. Methodol. , 267 (1996).

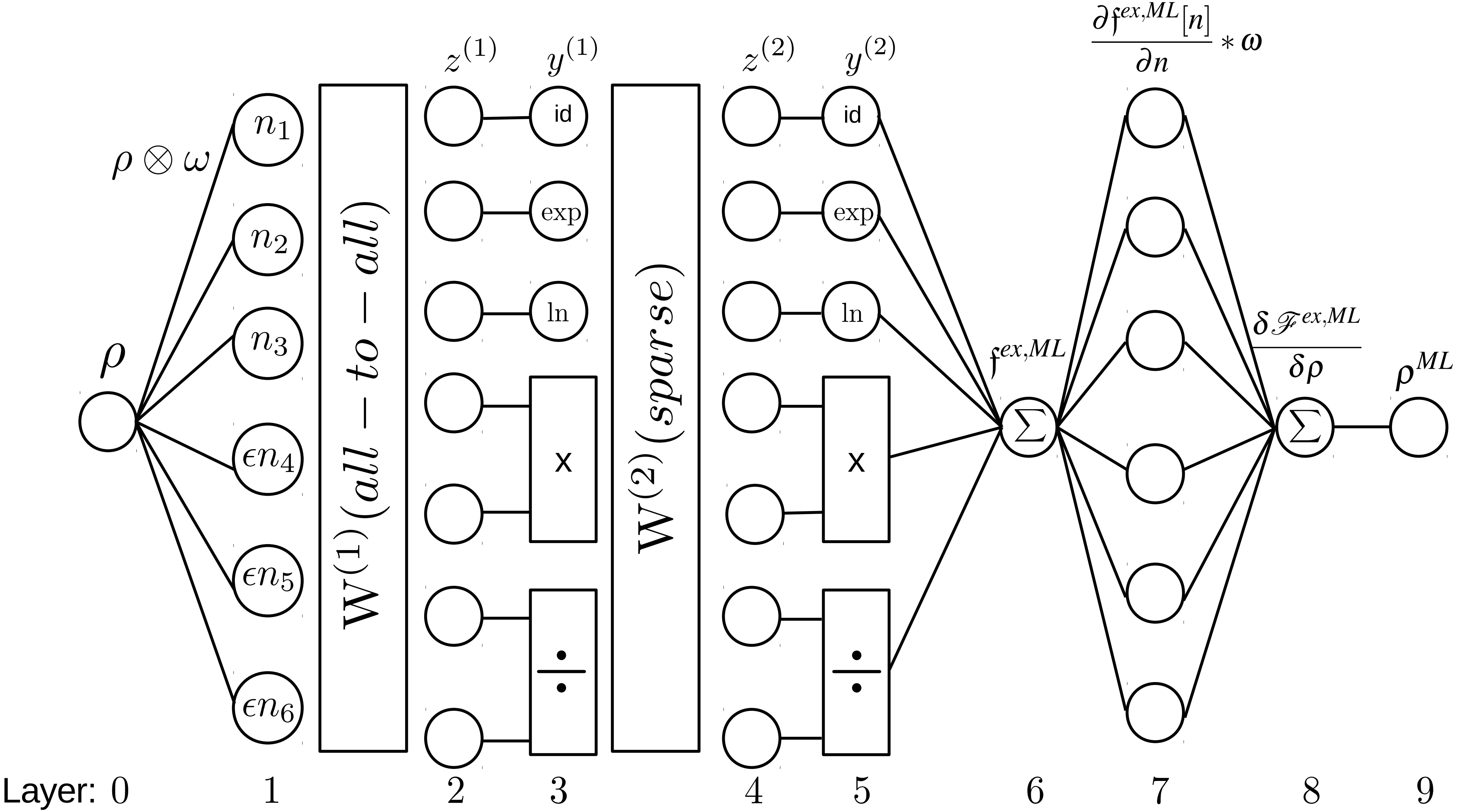
[18] Y. Rosenfeld, M. Schmidt, H. Löwen, and P. Tarazona, Physical Review E **55**, 4245 (1997).

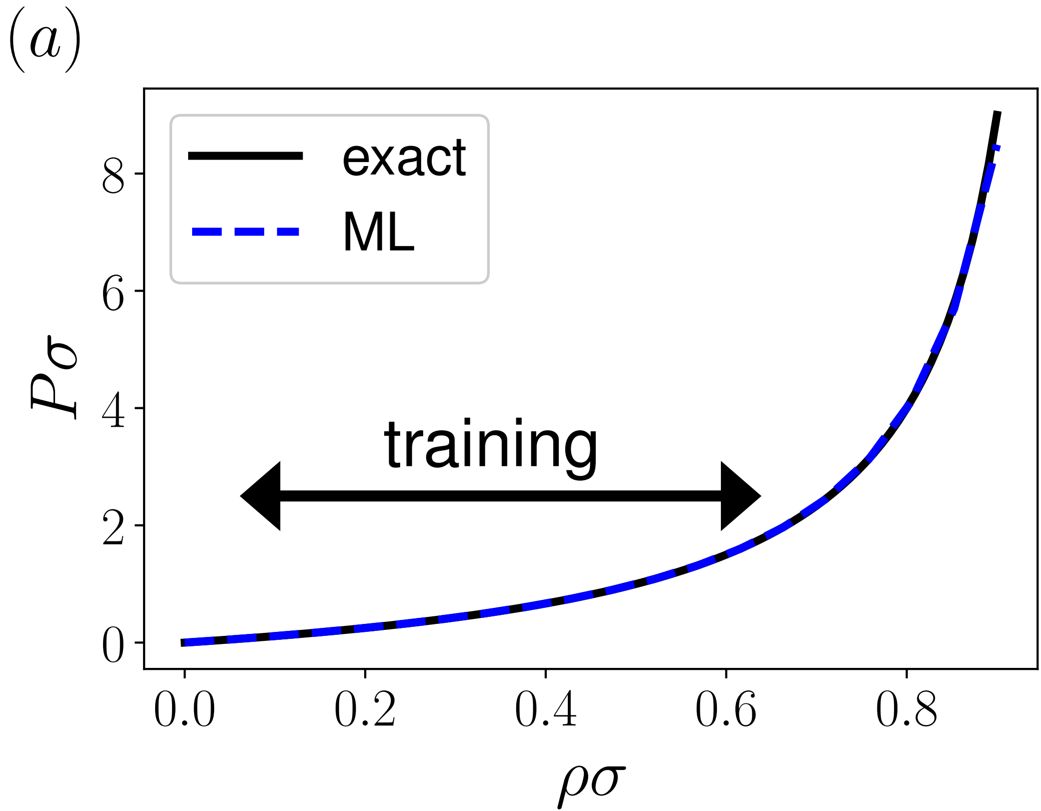
[19] J. A. Cuesta and A. Sánchez, J. Stat. Phys. **115**, 869 (2004).

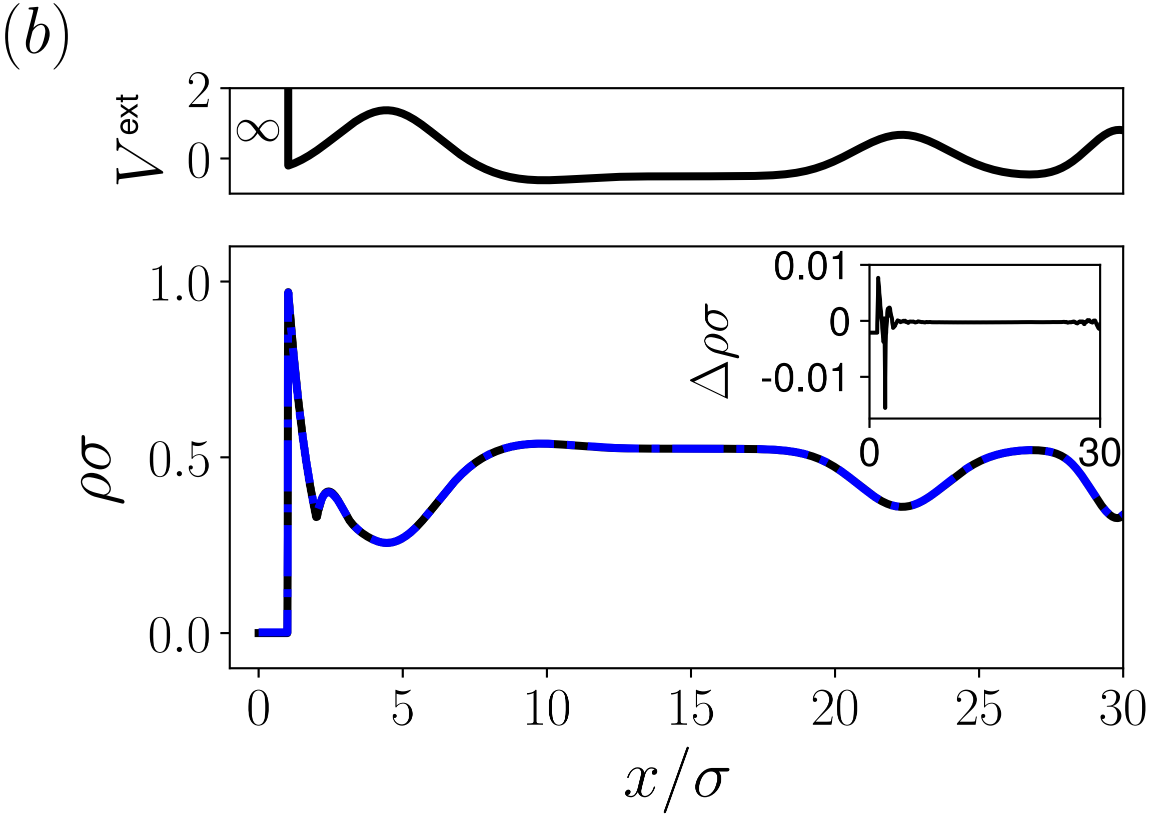
[20] V. P. Sergiievskyi, G. Jeanmairet, M. Levesque, and D. Borgis, J. Phys. Chem. Lett. **5**, 1935 (2014).

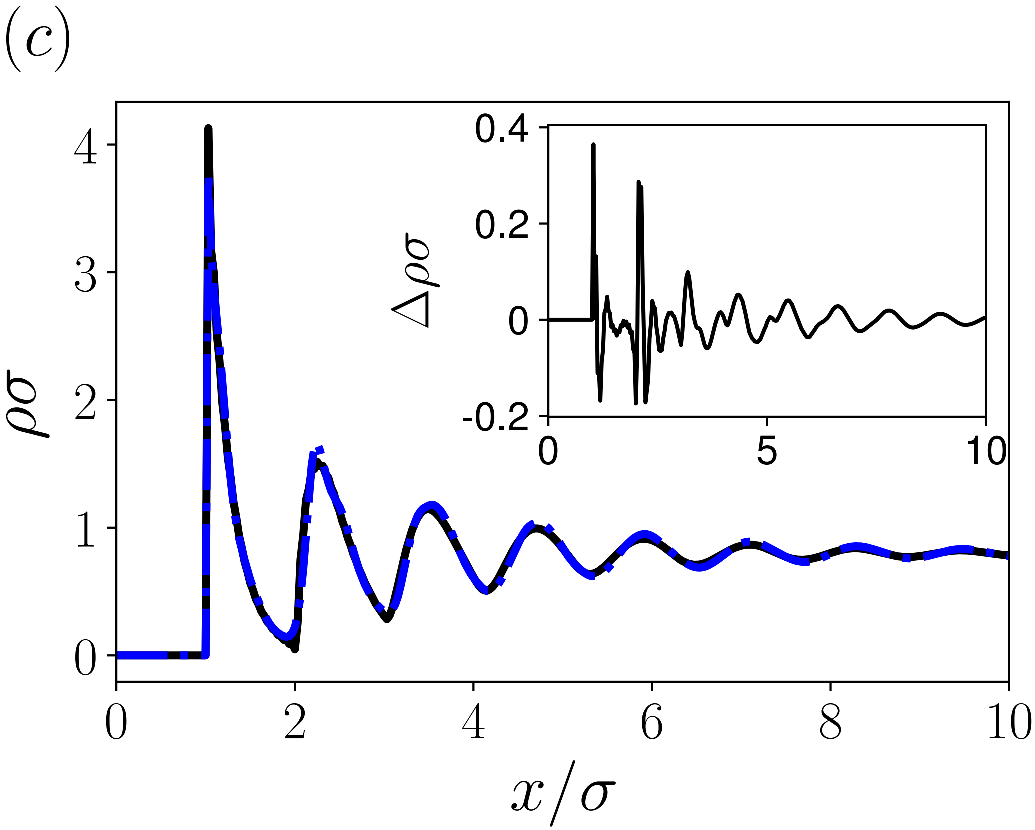
[21] G. Jeanmairet, M. Levesque, R. Vuilleumier, and D. Borgis, J. Phys. Chem. Lett. **4**, 619 (2013).

[22] G. Jeanmairet, M. Levesque, and D. Borgis, J. Chem. Phys. **139**, 154101 (2013).

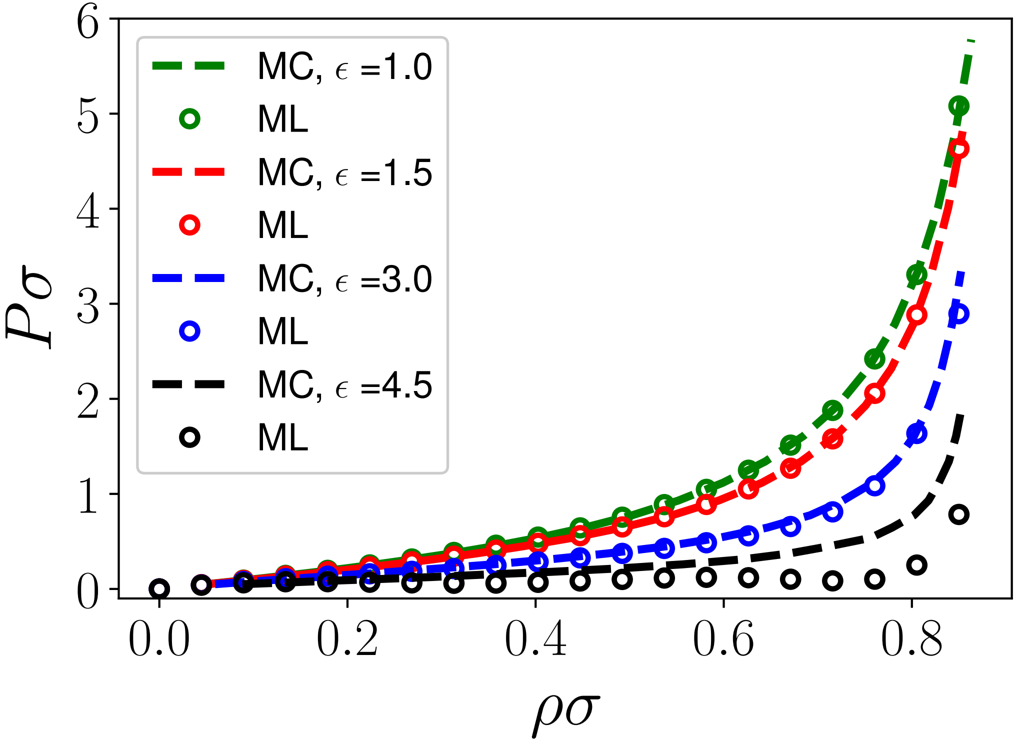

$\frac{\partial f^{ex,ML}[n]}{\partial n} * \omega$
$z^{(1)}$
$y^{(1)}$
$z^{(2)}$
$y^{(2)}$
$\rho \otimes \omega$
n_1
n_2
n_3
ϵn_4
ϵn_5
ϵn_6
ρ
$W^{(1)}(all-to-all)$
$W^{(2)}(sparse)$
id
exp
ln
x
÷
id
exp
ln
x
÷
$f^{ex,ML}$
Σ
$\frac{\delta \mathscr{F}^{ex,ML}}{\delta \rho}$
Σ
ρ^{ML}
Layer: 0
1
2
3
4
5
6
7
8
9


(a)
exact
ML
training
$P\sigma$
0
2
4
6
8
0.0
0.2
0.4
0.6
0.8
$\rho\sigma$

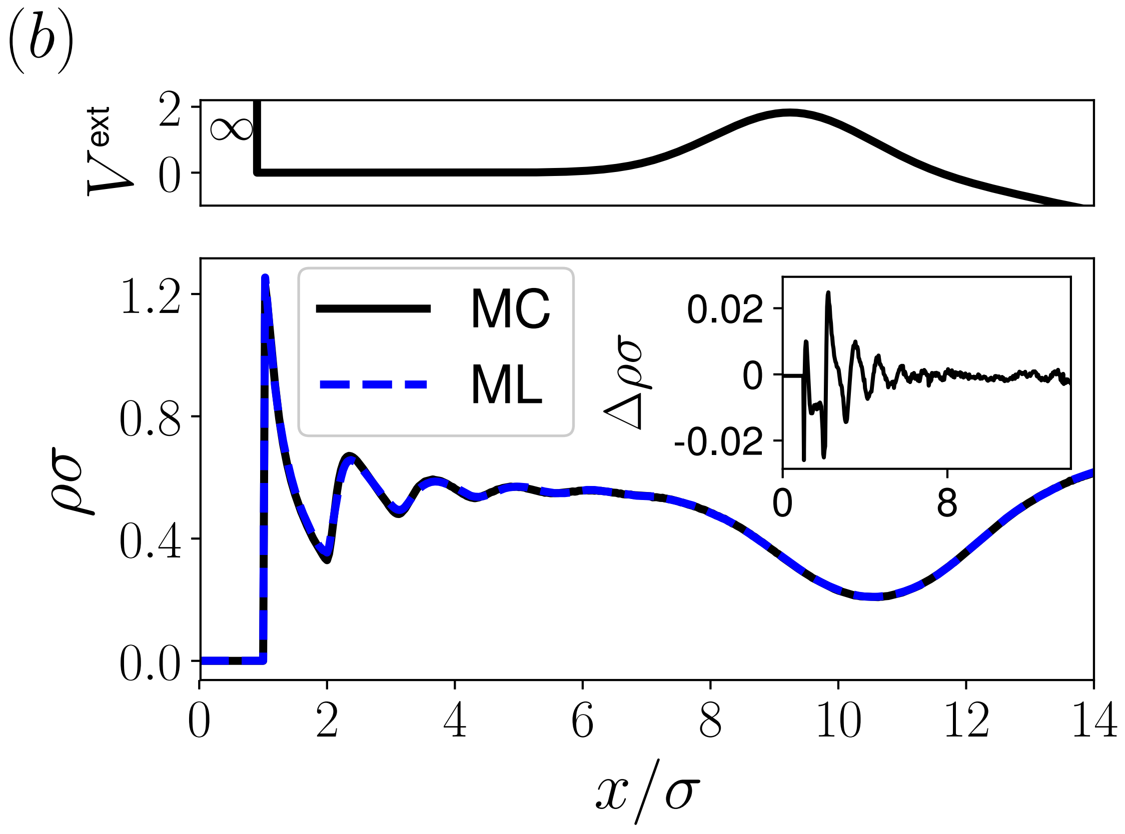

(b)
V^{ext}
2
0
∞
$\rho\sigma$
1.0
0.5
0.0
$\Delta\rho\sigma$
0.01
0
-0.01
0
30
0
5
10
15
20
25
30
x/σ

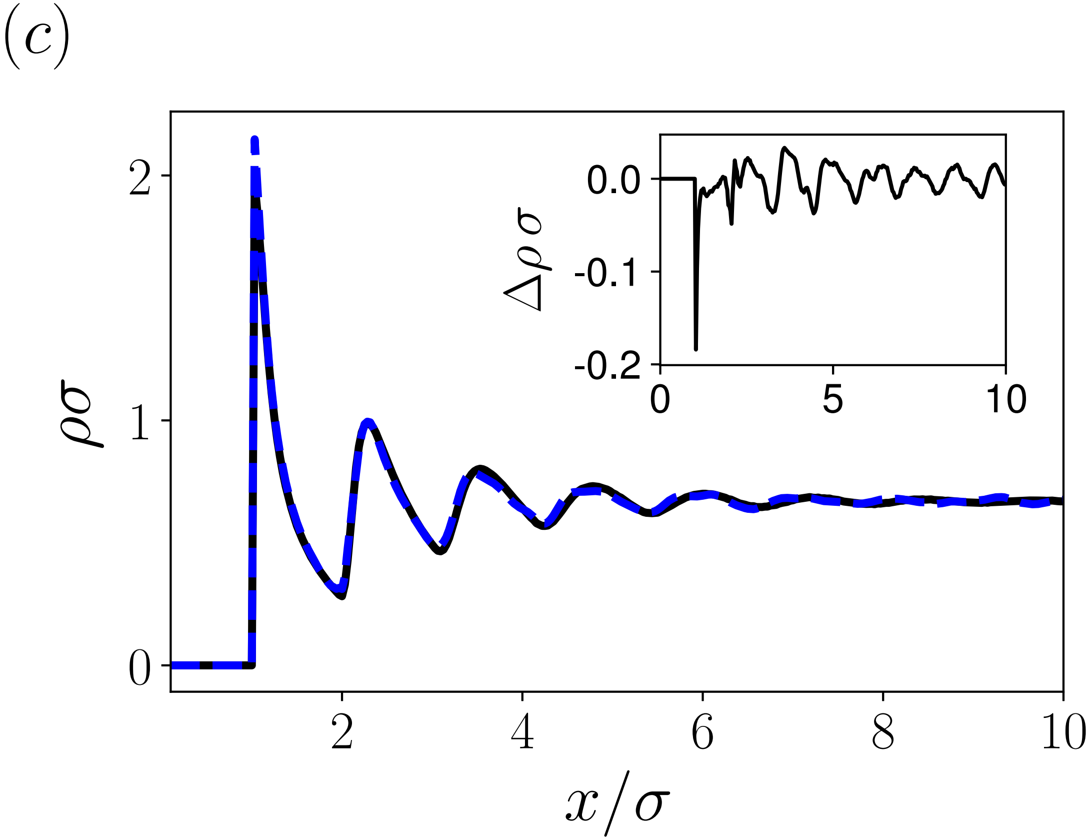

(c)
ρσ
0
1
2
3
4
x/σ
0
2
4
6
8
10
Δρσ
0.4
0.2
0
-0.2
0
5
10

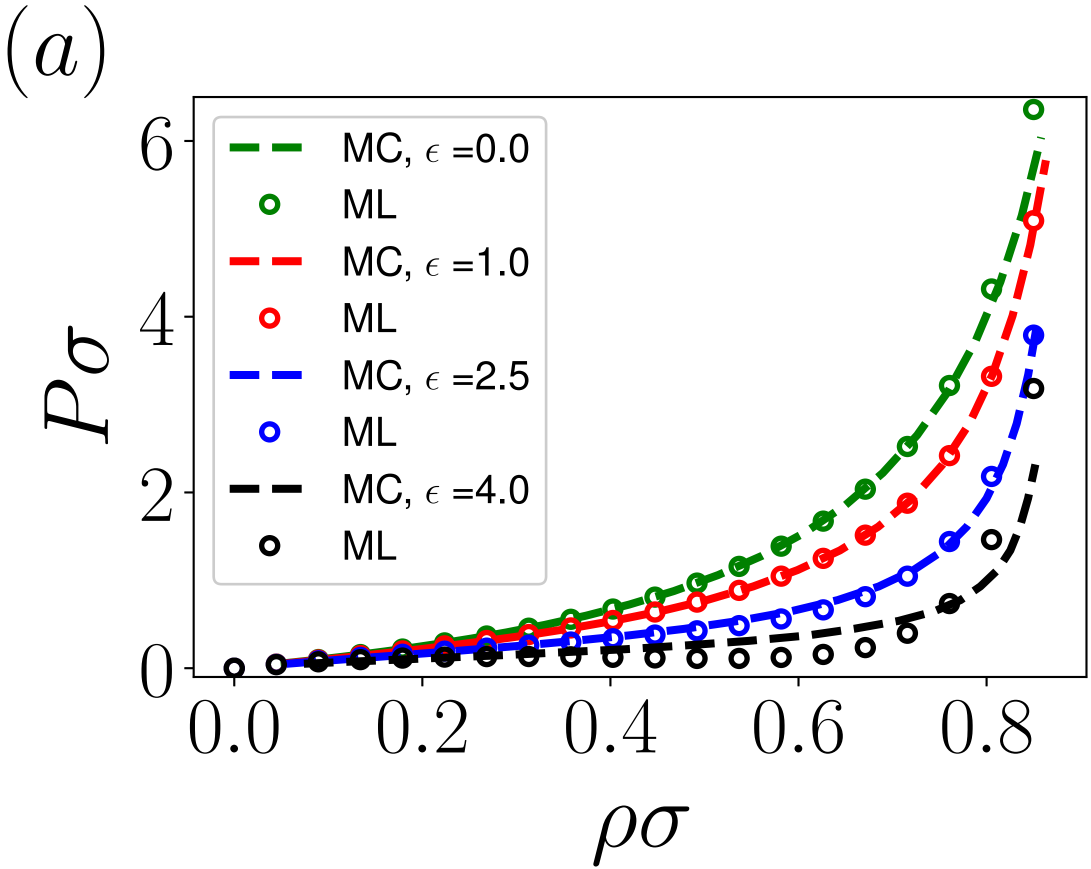
(a)



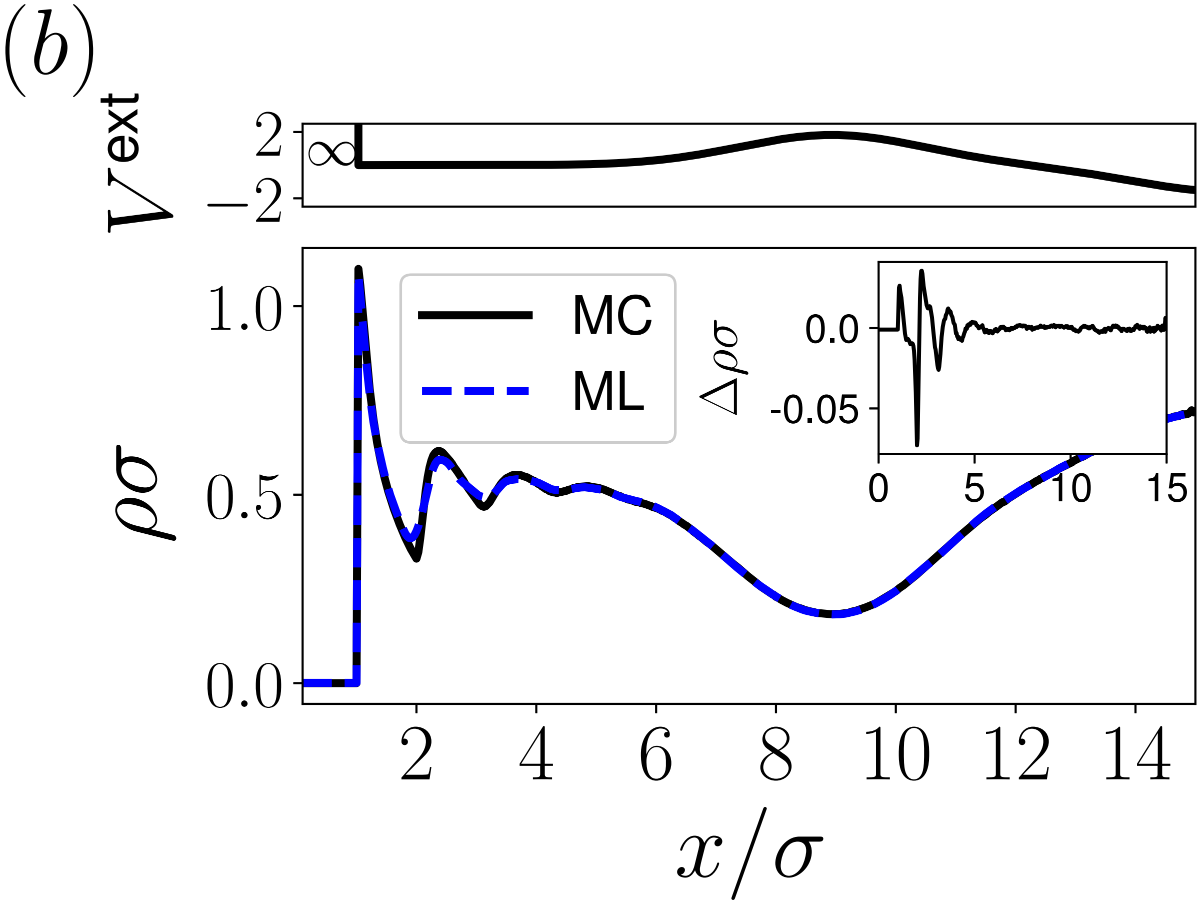
$P\sigma$
0
1
2
3
4
5
6
$\rho\sigma$
0.0
0.2
0.4
0.6
0.8
MC, ϵ =1.0
ML
MC, ϵ =1.5
ML
MC, ϵ =3.0
ML
MC, ϵ =4.5
ML

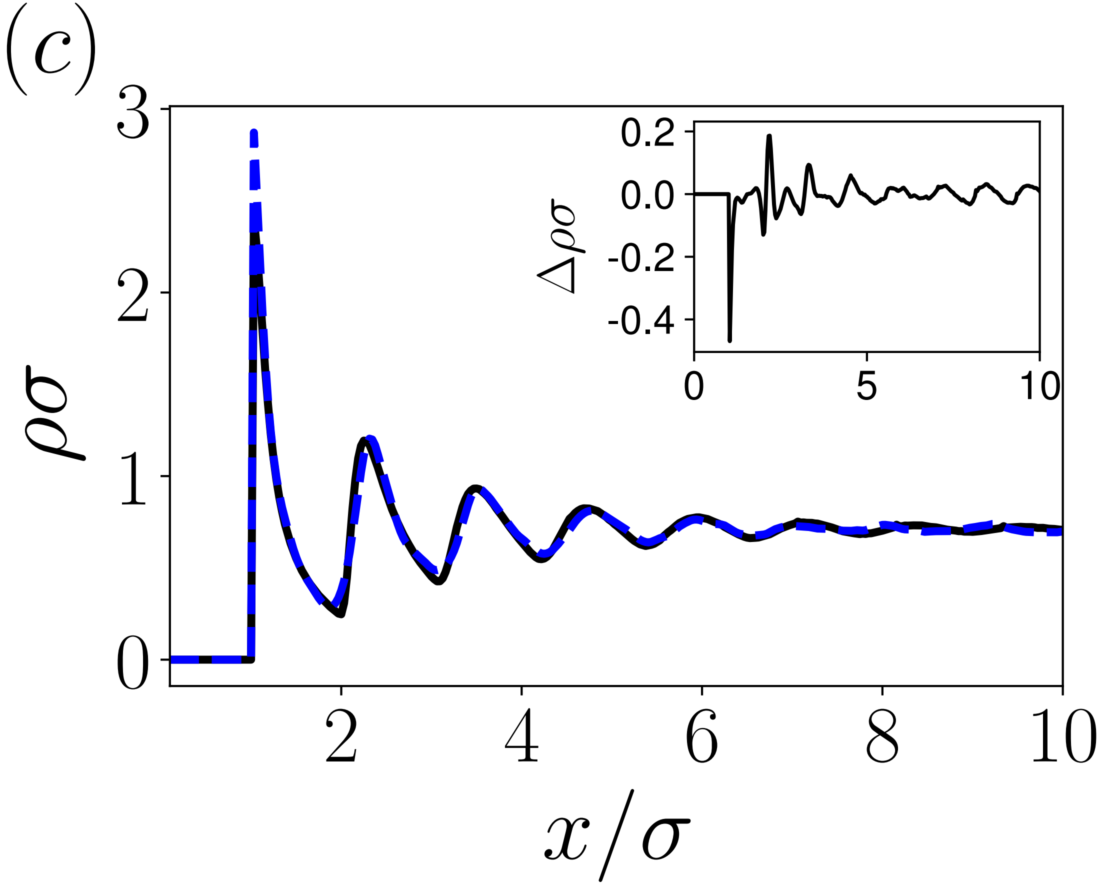

(b)
V^{ext}
2
0
∞
$\rho\sigma$
1.2
0.8
0.4
0.0
MC
ML
$\Delta\rho\sigma$
0.02
0
-0.02
0
8
0
2
4
6
8
10
12
14
x/σ


(c)
ρσ
0
1
2
x/σ
2
4
6
8
10
Δρσ
0.0
-0.1
-0.2
0
5
10

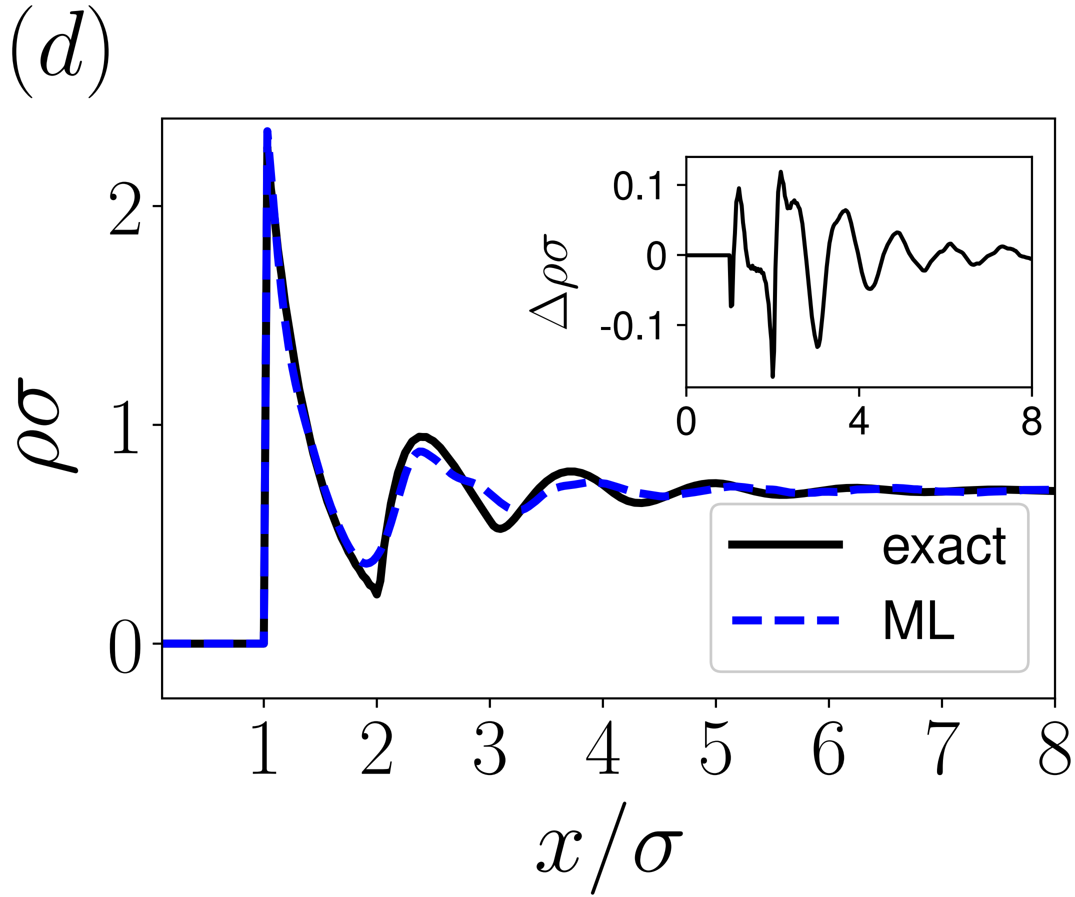


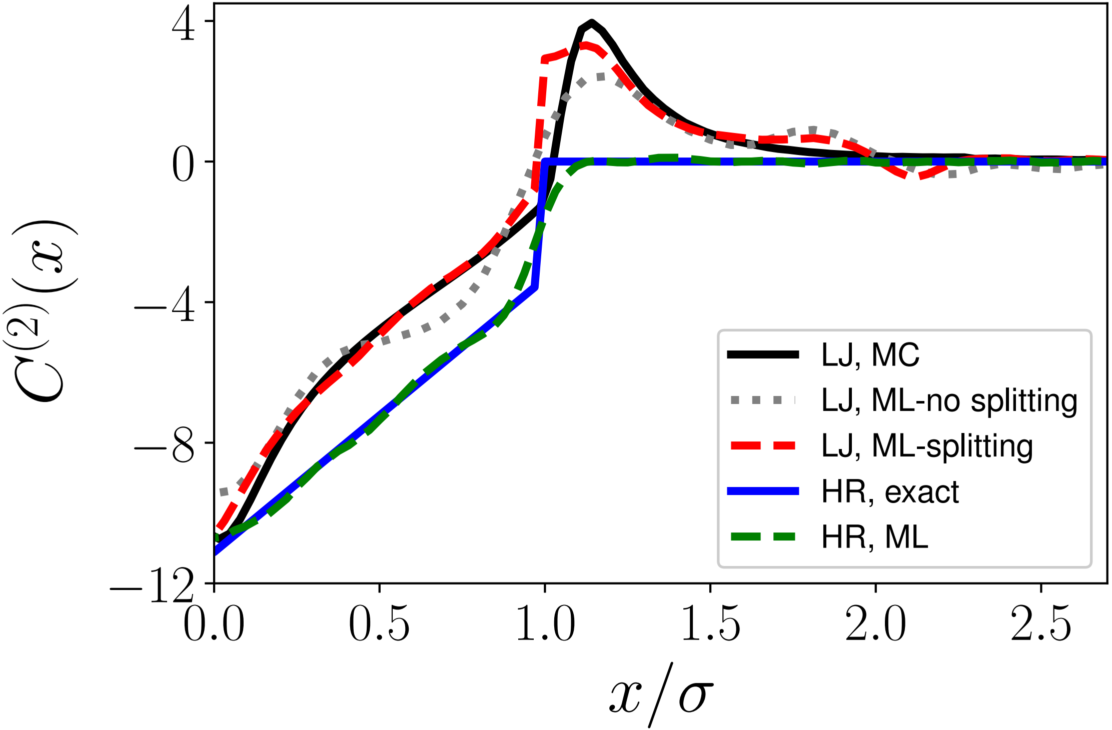
(a)
$P\sigma$
$\rho\sigma$
0
2
4
6
0.0
0.2
0.4
0.6
0.8
MC, ϵ =0.0
ML
MC, ϵ =1.0
ML
MC, ϵ =2.5
ML
MC, ϵ =4.0
ML


(b)
$V_{\rm ext}$
2
−2
∞
$\rho\sigma$
1.0
0.5
0.0
MC
ML
$\Delta\rho\sigma$
0.0
-0.05
0
5
10
15
2
4
6
8
10
12
14
x/σ



(c)
$\rho\sigma$
0
1
2
3
2
4
6
8
10
x/σ
$\Delta\rho\sigma$
0.2
0.0
-0.2
-0.4
0
5
10


(d)
ρσ
0
1
2
x/σ
1
2
3
4
5
6
7
8
Δρσ
0.1
0
-0.1
0
4
8
exact
ML


$C^{(2)}(x)$
4
0
-4
-8
-12
0.0
0.5
1.0
1.5
2.0
2.5
x/σ
LJ, MC
LJ, ML-no splitting
LJ, ML-splitting
HR, exact
HR, ML