Digital Discovery

PAPER

View Article Online
View Journal | View Issue

Check for updates

Cite this: *Digital Discovery*, 2025, **4**, 3578

Received 30th April 2025
Accepted 20th October 2025

DOI: 10.1039/d5dd00179j

rsc.li/digitaldiscovery

Generalized DeepONets for viscosity prediction using learned entropy scaling references

Maximiliam Fleck, *[a] Marcelle B. M. Spera, [a] Samir Darouich, [bc] Timo Klenk[a] and Niels Hansen *[a]

Data-driven approaches used to predict thermophysical properties benefit from physical constraints because the extrapolation behavior can be improved and the amount of training data be reduced. In the present work, the well-established entropy scaling approach is incorporated into a neural network architecture to predict the shear viscosity of a diverse set of pure fluids over a large temperature and pressure range. Instead of imposing a particular form of the reference entropy and reference shear viscosity, these properties are learned. The resulting architecture can be interpreted as two linked DeepONets with generalization capabilities.

1 Introduction

The scarcity of thermophysical property data presents a challenge in the development of new processes and materials, driving the long pursuit of predictive methods in chemical engineering.[1] For transport properties such as the shear viscosity, researchers have developed group contribution[2–7] and corresponding states methods[8–10] for pure substances. These methods may often yield inaccurate results, in particular for compounds not included in the training set – for which predictive errors can exceed 50%.[11,12] While molecular simulations offer an alternative approach for predicting transport properties (including viscosity), they face limitations in computational cost, transferability, and accuracy.[11–16] Machine learning approaches showed to be promising when sufficient training data are available, especially when enhanced by physics-based descriptors derived from molecular simulation.[17,18]

Considerable research has focused on exploiting the univariate relationship between a transport property and residual entropy, originally proposed by Rosenfeld for simple fluids.[19,20] If transport properties are rendered dimensionless through appropriate scaling references, they can be predicted across a wide temperature and pressure range. Entropy scaling approaches have been extensively applied to develop predictive models for both pure fluids[21,22] and fluid mixtures.[23,24] The univariate relationship between dimensionless shear viscosity and reduced residual entropy represents a dimensionality reduction that considerably improves the extrapolation behavior of data-driven approaches when combined with entropy scaling.[25]

While entropy scaling approaches differ in their details, *e.g.* in the way the transport property is made dimensionless,[26–32] they follow the same basic principles. In a preliminary work, we implemented an entropy scaling framework into a neural network architecture to predict the shear viscosity of pure fluids over a wide range of species and state points.[31] Molecules were represented by their PC-SAFT parameters. The Perturbed Chain Statistical Associating Fluid Theory (PC-SAFT) equation of state[33] uses only a few substance-specific parameters and we were able to validate that they can serve as highly effective molecular descriptors in a machine-learning context and are of low dimensionality compared to other molecular fingerprints commonly employed.[34–36]

In our preliminary work,[31] we adopted the Chapman–Enskog scaling reference – which provides a meaningful low-density behavior. However, alternative scaling relations may be more appropriate for dense states. In the present work, we generalize the previous approach by allowing the neural network to learn the optimal scaling relation during training, rather than imposing a specific scaling model *a priori*. This results in an architecture that can be interpreted as two linked DeepONets with generalization capabilities (GenDeepONets), with application cases beyond the application discussed in this work. Compared to the previous architecture,[31] it demonstrates superior performance metrics on a more challenging dataset and has undergone a more thorough evaluation analysis, establishing our DeepESNet (Deep Entropy Scaling Network) as a significant advancement over the previous methodology and other feed forward methods.

[a] Institute of Thermodynamics and Thermal Process Engineering, University of Stuttgart, Pfaffenwaldring 9, 70569 Stuttgart, Germany. E-mail: maxi_fleck@posteo.com; hansen@itt.uni-stuttgart.de

[b] Institute for Artificial Intelligence, University of Stuttgart, Universitätsstraße 32, 70569 Stuttgart, Germany

[c] Institute for Theoretical Chemistry, University of Stuttgart, Pfaffenwaldring 55, 70569 Stuttgart, Germany

2 Entropy scaling

As shown in Fig. 1, the behavior of viscosities in the temperature–pressure (T, p) or temperature–density (T, ρ) space is rather intricate with a non-trivial property surface, where phase transitions lead to discontinuities.[26,31] Rosenfeld[19,20] demonstrated that transport coefficients, including viscosity, thermal conductivity, and self-diffusion coefficients, can be approximated as univariate functions solely dependent on reduced residual entropy, provided that these transport coefficients are defined as dimensionless quantities in suitable manners. In the residual entropy space, phase transitions no longer manifest as discontinuities. This greatly simplifies the problem of developing predictive methods, as two-phase systems or domains of large curvatures on the surface $\eta(T, p)$ do not have to be described.

Rosenfeld initially proposed the entropy scaling approach for simple fluids. However, follow up studies showed the applicability of this principle to strongly non-spherical,[37–45] polar, and hydrogen-bonding fluids, including water.[21,23,26,46,47] It has been demonstrated that the approach is applicable to a wide variety of substance groups.[23,26] Entropy scaling can be integrated into a neural network architecture to reliably predict viscosities for many substances and state points as shown in our preliminary work.[31] In this work, we want to generalize this approach.

The reduced residual entropy $s^*(T, \rho)$ at each chosen state point is computed using the PC-SAFT equation of state[33] as

$$s^*(T,\ \rho) = \frac{s(T,\ \rho) - s^{\mathrm{ig}}(T,\ \rho)}{N_{\mathrm{A}} k_{\mathrm{B}} \zeta_s} = \frac{s^{\mathrm{res}}}{\zeta_s} \tag{1}$$

with s^{res} the residual entropy, $s(T, \rho)$ the substance entropy, and $s^{\mathrm{ig}}(T, \rho)$ the entropy of the ideal gas at the same temperature and density, Avogadro's constant N_{A}, Boltzmann's constant k_{B}, and normalizer ζ_s which is defined differently in different entropy scaling approaches.[26–31] Typically, ζ_s is a substance-specific constant parameter related to the size of the molecule.

To establish a dimensionless measure of viscosity, $\eta^* = \eta/\eta^{\mathrm{ref}}$ in a suitable manner, a reference viscosity, for example the Chapman–Enskog[48,49] viscosity, is introduced. Typically, the reference viscosity is a function of absolute or reduced temperature and substance-specific. Depending on the chosen approach, it can also rely on additional inputs such as entropy. In this work, the reference will be learned. The aim of the reference is to minimize the noise. Therefore, we can analyze and compare different approaches due to their potential to minimize noise in the entropy space, here referred to as denoising. Our findings indicate that the reference viscosity is fulfilling its denoising function when dependent on temperature and substance-specific parameters:

$$\ln(\eta^{\mathrm{ref}}) = f(T, \mathbf{x}^{\mathrm{ref}}) \tag{2}$$

with f as a function of temperature T and species dependent features $\mathbf{x}$. The dimensionless viscosity leads to coherent entropy scaling behavior and is typically utilized in logarithmic form,

$$\ln(\eta^*) = \ln(\eta/\eta^{\mathrm{ref}}) = f(s^*, \mathbf{x}^*) \tag{3}$$

with f as a function of entropy s^* and species dependent features $\mathbf{x}^*$. $f(s^*, \mathbf{x}^*)$ can be a polynomial or completely or partly learned.

3 Generalized neural network architecture

First, we assume that species dependent features are the same throughout the model, *i.e.*, $\mathbf{x}^{\mathrm{ref}} = \mathbf{x}^* = \mathbf{x}$. Reformulating eqn (2) and (3):

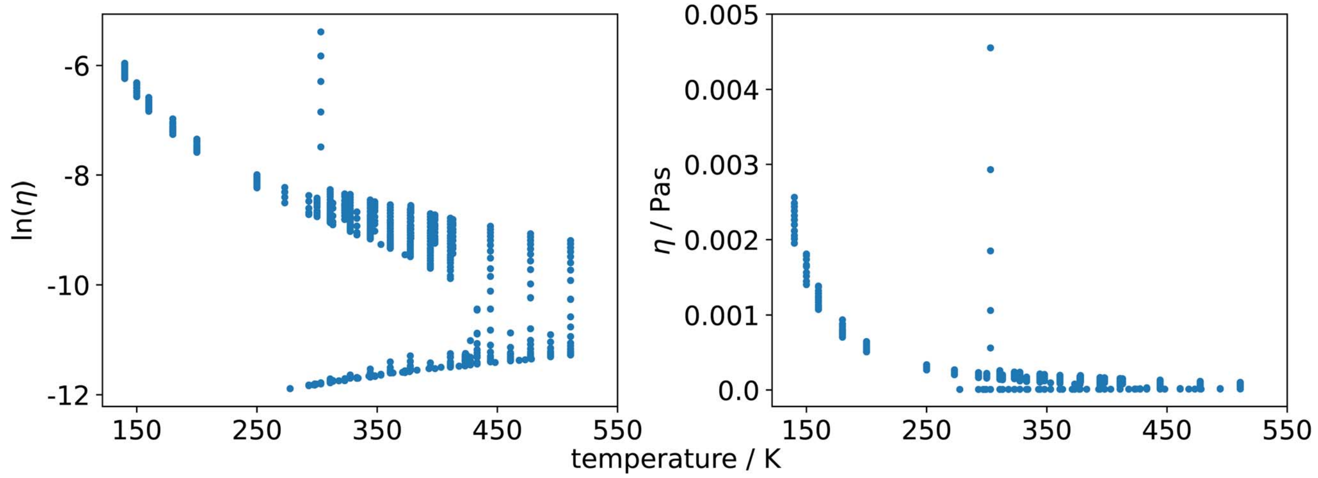

Fig. 1 Viscosity and log-transformed viscosity of butane plotted over temperature. Differing viscosities at one temperature indicate different state points (T, p) and (T, ρ), respectively. The phase behavior is visible in the left plot. Vapor state points (low pressure/low density) are clearly separated from the liquid phase (high pressure/high density). At high temperatures, the critical region connecting the two phases also becomes visible. It is evident that predicting viscosity is very challenging, as values even of only one specie span several magnitudes and the phase behavior is not trivial.

$$\ln(\eta) = \ln(\eta^{\text{ref}}) + \ln(\eta^*) = f(T, \mathbf{x}) + f(s^*, \mathbf{x}) \tag{4}$$

with species dependent features $\mathbf{x}$, which are a model input together with $f(s^*, \mathbf{x})$ and T.

Now we can further investigate the f functions. Starting with $f(s^*, \mathbf{x})$, which is typically a polynomial with substance-specific parameters and normalized entropies. Therefore, we define $f(s^*, \mathbf{x})$ as the dot product of a substance-specific parameter vector and a residual entropy feature vector

$$\ln(\eta^*) = f(s^*, \mathbf{x}) = \mathbf{p}^*(\mathbf{x}) \times \mathbf{s}^*(s^*(\mathbf{x})) \tag{5}$$

with the substance-specific model parameter vector $\mathbf{p}^*$ and entropy feature vector $\mathbf{s}^*(s^*(\mathbf{x}))$. The latter is substance-dependent through $\zeta_s(\mathbf{x})$ and can be rewritten as

$$s^* = \frac{s^{\text{res}}}{\zeta_s(\mathbf{x})} \tag{6}$$

with the substance-specific temperature parameter $\zeta_s(\mathbf{x})$ and residual entropy $s^{\text{res}}(T, \rho)$, which is a model input and can be computed using an equation of state within pre-processing.

We can do the same for our reference viscosity and define

$$\ln(\eta^{\text{ref}}) = f(T, \mathbf{x}) = \mathbf{p}^{\text{ref}}(\mathbf{x}) \times \mathbf{T}^*(T^*(T, \mathbf{x})) \tag{7}$$

with a substance-specific reference parameter vector $\mathbf{p}^{\text{ref}}(\mathbf{x})$ and a temperature feature vector $\mathbf{T}^*(T^*(T, \mathbf{x}))$, which is a function of the reduced temperature

$$T^* = \frac{T}{\zeta_T(\mathbf{x})} \tag{8}$$

with the substance-specific temperature parameter $\zeta_T(\mathbf{x})$ and temperature T, which is a model input.

The equations can be translated into the architecture shown in Fig. 2. We call this architecture DeepESNet, which refers to two generalized DeepONets that are connected to calculate viscosity through entropy scaling. Each block has the general structure shown in Fig. 3. The model inputs $\mathbf{x}$, T, and s^{res} are used for different neural networks. On the reference GenDeepONet block, the branch network computes the reference

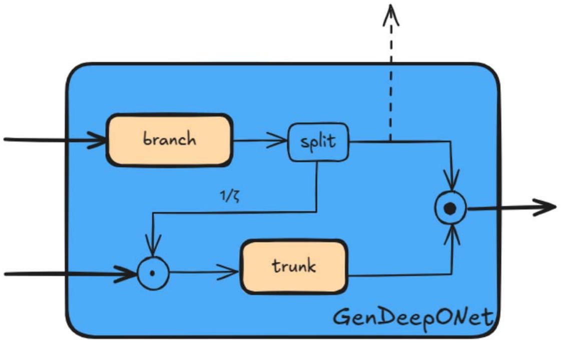

Fig. 3 DeepES network unit described with the naming from DeepONet using the terms branch and trunk network. The dotted line indicates that we can pull out additional information which might be useful in some cases.

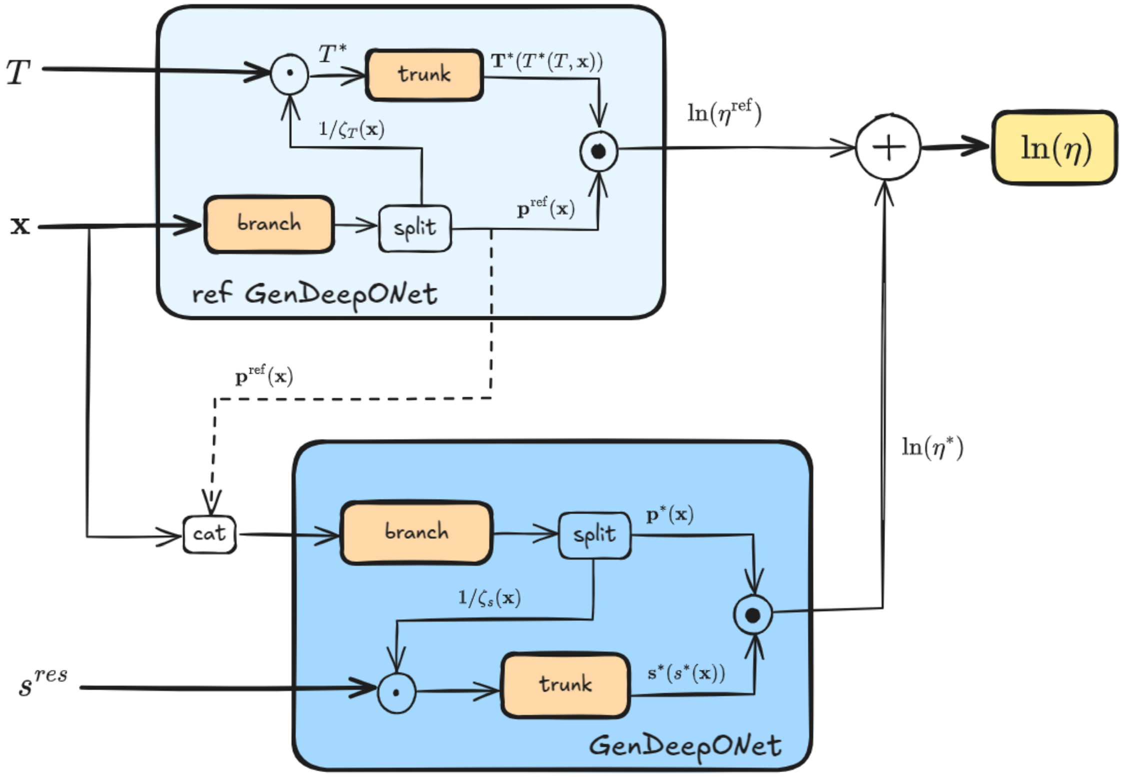

Fig. 2 DeepESNet: architecture to predict viscosities. Branch and trunk are fully connected deep neural networks using ReLu activation.[50] The architecture is translated from eqn (4)–(8), with PC-SAFT parameters as species dependent features $\mathbf{x}$.

parameters $\mathbf{p}^{\text{ref}}(\mathbf{x})$ and $\zeta_T(\mathbf{x})$ as a function of $\mathbf{x}$. The substance-specific parameter $\zeta_T(\mathbf{x})$ is then used to reduce the temperature. The reduced temperature is the input of the trunk network which computes the temperature feature of the reference as a function of the reduced temperature T^*. With that, we have addressed eqn (7) and (8).

On the second GenDeepONet block, the branch network receives as input the model parameters from the reference state, $\mathbf{p}^{\text{ref}}(\mathbf{x})$, which come from the reference GenDeepONet block and are combined with input $\mathbf{x}$ to compute the model parameters $\mathbf{p}^*(\mathbf{x})$ and $\zeta_s(\mathbf{x})$ as a function of $\mathbf{x}$. The trunk network computes the entropy features as a function of the reduced residual entropy s^*. This, in turn, addresses eqn (5) and (6). We have now the information required to obtain the viscosity through eqn (4). The branch networks can be considered separately, combined, or interconnected. We found that the interconnected approach passing $\mathbf{p}^{\text{ref}}(\mathbf{x})$ enables good prediction. Nevertheless, we provide on the SI a benchmark against other architectures.

4 Comparison to DeepONet framework

The formalism derived from the entropy scaling principle and the corresponding architecture bear strong structural similarities to interconnected DeepONets.[51,52] DeepONets have emerged as promising surrogate solvers for partial differential equations (PDEs), characterized by the following integral form:

$$\mathrm{Gu}(t) = x(0) + \int_0^t f(\mathrm{Gu}(\tau), u(\tau), \tau)\mathrm{d}\tau \tag{9}$$

A DeepONet consists of two sub-networks: a branch network and a trunk network operating on different function spaces. The branch network's function space is mapped to the trunk network's function space. The trunk network receives inputs defining the desired output location, which for PDEs typically represent spatial or temporal coordinates. In our work, the inputs to our trunk networks are temperature and residual entropy. The branch network receives inputs characterizing the molecule ($\mathbf{x}$). This approach maps a function space describing molecular characteristics to the entropy function space, analogous to how DeepONet maps from input function space to solution function space.

$$\mathrm{Gu}(s) = \eta(s) = \eta(0) + f(\eta(s), \mathbf{x}, s(T, \rho)) \tag{10}$$

Which can be related to eqn (4) with $\eta(0) = \eta^{\text{ref}}$. In our case, $\eta(0)$ is unknown and needs to be learned. Therefore, we end up with two interconnected DeepONets. Our architecture extends beyond the standard DeepONet framework through an idea that might be transferable to PDE surrogate solvers. Our trunk network utilizes normalized or generalized inputs, where the normalization parameters are learned by the branch network. In other words, the inputs of the trunk networks function space are normalized based on the branch networks function space. This approach aligns with established thermodynamic principles—absolute temperatures often yield less insight than temperatures normalized relative to critical points, while in entropy scaling, entropies are typically normalized by parameters representing molecular size to enhance generalization.

This normalization strategy has potential applications beyond our specific domain and could benefit PDE surrogate solvers more broadly. Transport processes, such as the diffusion-reaction systems examined in the original DeepONet literature, often exhibit similar behavioral patterns but operate across different time scales, particularly when multiple terms in the governing equations vary simultaneously. Therefore, learning appropriate generalizations of the time axis based on driving forces represents a promising direction for improving surrogate model performance and transferability.

5 PC-SAFT representation parameters

We chose PC-SAFT parameters as our species dependent features $\mathbf{x}$. Previous work demonstrated the suitability of PC-SAFT parameters and the molecular mass for machine learning applications.[31,36] In the PC-SAFT equation of state, molecules are conceptualized as chains composed of spherical segments. In the applied version, all segments are of uniform size, and branches and rings are not explicitly represented in the model. The underlying potential is a Lennard–Jones potential. This provides three parameters (m, σ, ε) for chained molecules, such as n-alkanes. In PC-SAFT, m represents the effective (non-integer) number of segments per molecule, σ is the segment size parameter, and ε is the energy parameter of the intermolecular potential per segment. Three further parameters are required to adequately describe hydrogen-bonding (associating) groups (κ, ε_{AB}) and dipolar (μ) molecules. Here, κ is the effective association volume parameter, ε_{AB} the association energy between association sites A and B parameter, and μ dipole moment parameter. In this work, we rely on PC-SAFT parameters obtained from SMILES strings with transformers from Winter *et al.*[53]

6 Data and training

The models are trained using viscosities obtained from the Dortmund Data Bank, DIPPR and ThermoML.[54–56] Certain substances and families are overrepresented in the data set. However, given that these substances and families are of significant importance, particularly within chemical families such as alkanes and alcohols, this overrepresentation can be considered advantageous. Furthermore, it should be noted that the data underwent a relatively brief cleaning process, which primarily served to remove non-plausible data. The training process was sufficiently robust, and there was no indication that a more elaborate outlier detection method was required.

As previously mentioned, the PC-SAFT equation of state was developed to describe molecules as chains of spherical segments. Therefore, only a few molecular families comprising cyclic molecules were not utilized in the training of the final model. Nevertheless, we left cyclical and polycyclical families like naphtalenes in the dataset, in order to examine and demonstrate the representation's limitations.

First, we split the data at family and species level. Small molecules are considered to be much more difficult to predict and yet carry important information.[53] Therefore, species with a molecular weight of less than 68 $\frac{\text{g}}{\text{mol}}$ and a SMILES string length smaller than 5 symbols were incorporated into the training set. For families with more than 12 species in the dataset, we used a training, validation, and test split of 70%/15%/15% at species level. For families with less than 12 and more than 9 species, we used a training and validation split of 85%/15%. For families with less than 9 species, we used all species for training. From a total of 729 different species, we end up with 115 species that are part of the validation data but not the training data, and 112 species that are part of the test data but not the training data. The validation and test data were then populated with samples from the training set, resulting in a final training, validation, and test split of 50%/25%/25%. The total number of samples is 76 915. This train/test/val split is denoted as 50/25(15)/25(15) split, with 15% being the amount of data not present at the training set. Further details on the dataset split can be found in the SI. We also want to point out that the datasets largest family is the family with the label unknown, consisting of species that were not assigned to a family. Hence, the species in validation and test sets that are not part of the training set are heterogeneous.

Besides the dimensions of the branch and trunk networks, the size of the output embeddings of the networks are treated as hyperparameters. We used the Adam optimizer[57] with L2 (mean squared) loss and treated regularization penalty,[58] learning rate, and batch size as hyperparameters. More implementation details are available in the SI.

7 Results

Here we discuss the temperature dependency of the reference viscosity, $\ln(\eta^{\text{ref}})$, and show results of a trained model utilizing the full architecture. We also discuss further utilization options. Deviations between model and experiment are given as mean absolute relative deviations in percent (MARD-%), unless specified otherwise. Another measure used is the median absolute relative deviations in percent (median ARD-%).

$$\text{MARD} = \frac{100\%}{N}\sum_{i=1}^{N}\frac{\left|\eta_{\text{pred},i} - \eta_{\text{exp},i}\right|}{\eta_{\text{exp},i}} \tag{11}$$

$$\text{ARD} = 100\%\frac{\left|\eta_{\text{pred},i} - \eta_{\text{exp},i}\right|}{\eta_{\text{exp},i}} \tag{12}$$

7.1 Temperature dependency of the reference viscosity

The reference viscosity is responsible for the denoising of residual entropy space, and is a function of both substance-specific parameters and temperature. In this section, we will validate the last affirmation, *i.e.*, if the reference viscosity can minimize noise in the entropy space only as a function of **x** and *T*. This is required since alternative approaches incorporate additional state-point-specific dependencies.[26–31,36] The ability of the reference viscosity to denoise the data – whether derived analytically or through machine learning – is crucial. Due to the separation of the functional areas of the reference and the actual prediction, the latter depends on the noise reduction in the input data. When this is not achieved, it results in a considerable dependency not only on the training set, but also on the state points. This can lead to problems with imbalanced datasets. To test whether a temperature-dependent reference is sufficient, we employed only the components of the architecture illustrated in Fig. 2 that pertain to the reference viscosity. This means that the reference GenDeepONet is used to predict $\ln(\eta^{\text{ref}})$ whereas the second GenDeepONet is not used. We will refer to it as denoising architecture.

An alternative loss function was delineated for the purpose of training only the branch and trunk networks from the reference GenDeepONet. According to eqn (4), subtracting $\ln(\eta^{\text{ref}})$ from experimental $\ln(\eta)$ gives us $\ln(\eta^*)$, which should be noise-free when plotted over the residual entropy if the reference viscosity is predicted properly. A noise-free correlation between $\ln(\eta^*)$ and the residual entropy can be approximated with a high degree of accuracy by a polynomial. The L2 (mean squared) loss between a polynomial fit and $\ln(\eta^*)$ is therefore defined as the loss function.

We used all available data from 16 *n*-alcohols for training. Denoising results for octan-1-ol is shown in Fig. 4, where it is possible to see that our reference GenDeepONet was able to minimize the noise in the viscosity data (left plot), leading to a less spread dataset – and a more accurate prediction (right plot). Results for butan-1-ol, ethanol, and octan-1-ol are shown in Fig. 5. The combination of temperature-dependent denoising with polynomial fitting demonstrates excellent agreement with Experimental data. This validates our hypothesis that the reference viscosity depends only on substance-specific parameters **x** and temperature, eliminating the need for additional state point dependencies such as density or entropy. For *n*-alcohols, this approach achieves better agreement with experimental results compared to classical entropy scaling methods:[23,25,26] for octan-1-ol, for example, we achieved 4.35% MARD while the value reported for a substance-specific fitting through an analytical equation of state was 5.13%.[23] Although the present dataset is challenging, only a temperature-dependent reference viscosity was sufficient to reduce the noise. Non-denoised data could lead to instability in training and unpredictable fitting behavior especially in such heterogeneous datasets. The methodology employed can be utilized to construct highly accurate models of individual substances, groups of substances and, possibly, mixtures.

As a final remark, we would like to mention that the polynomial approach for investigating the reference viscosity can be substituted with a Gaussian process, where optimization would aim the marginal log likelihood. This alternative framework effectively minimizes entropy within the data itself. Consequently, this methodology can be combined with classical entropy scaling approaches to optimize equation of state parameters by reducing noise in the data. The possibility of

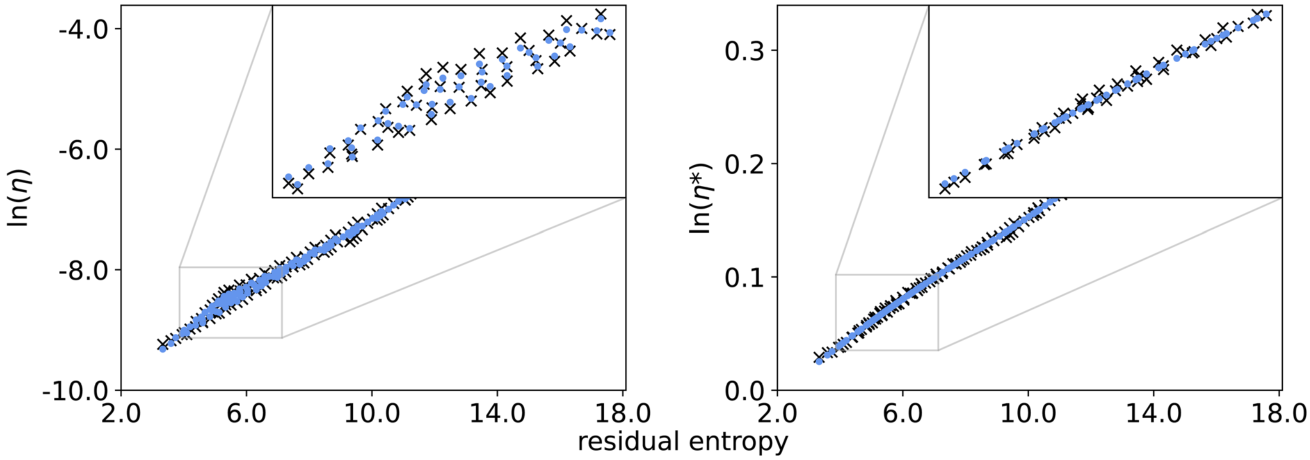

Fig. 4 Denoising results for octan-1-ol. Predictions made on the viscosity data (ln η) are plotted on the left and show at first glance a good agreement with experimental results (black crosses). Nevertheless, one must be aware of the scale and the fact that the data is scattered through approx. 3 orders of magnitude. The effect of the denoising architecture is visible on the right, where predictions made on the reduced viscosity (ln η*) are much more accurate and the entire dataset has the same order of magnitude throughout the entropy space.

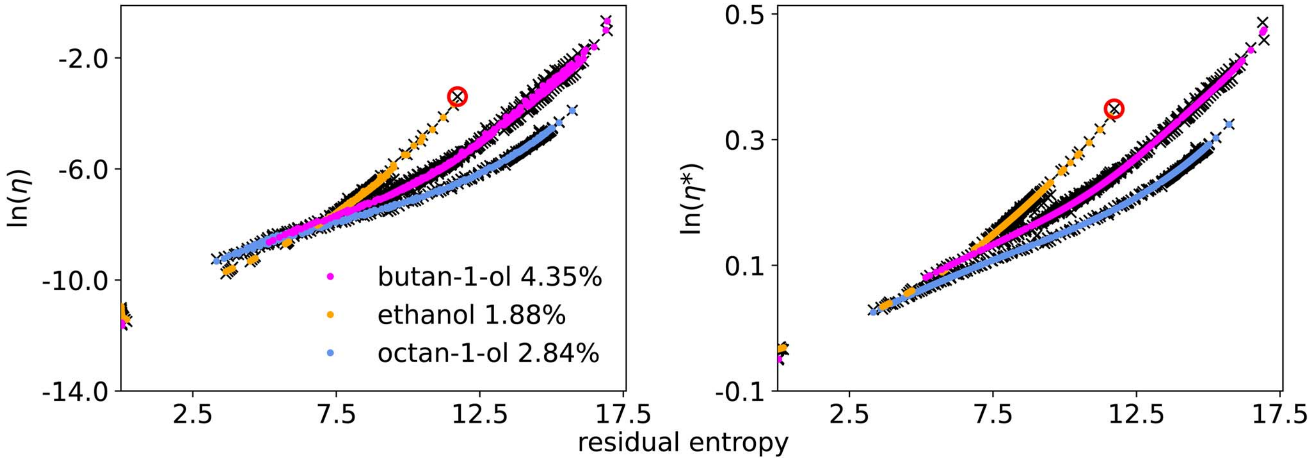

Fig. 5 Denoising results for a few selected n-alcohols, with MARD shown next to the substance name. Predictions made on the viscosity data (ln η) are plotted on the left and have visible good agreement with experimental results (black crosses). The effect of the denoising architecture is visible on the right (ln η*). It is important to highlight that due to entropy scaling, the approach is resilient to overfitting outliers: the red-circled black cross is part of the butan-1-ol experimental data but lies together with ethanol data points. Nevertheless, the denoising procedure for butan-1-ol was not influenced by this outlier. For comparison, the MARD for butan-1-ol, ethanol, and octan-1-ol from classical entropy scaling with substance-specific fitting is reported as 5.79%, 3.40%, and 5.13%.[23]

evaluating the reference viscosity directly also opens up a wide range of applications for developing theories on transport properties.

7.2 Trained full model architecture

The full model architecture was trained using the training data, and the hyperparameters were optimized using the validation data. We trained the model on different 50/25(15)/25(15) splits. The main investigation and benchmarks were conducted on a challenging 50/25(15)/25(15) split (Split 1 from Table 1). For training data, the MARD between model prediction and experimental data is 6.05% and the median ARD is 4.02%. For validation data, the MARD between model prediction and experimental data is 7.59% and the median ARD is 4.20%. For test data, the MARD between model prediction and experimental data is 10% and the median ARD is 4.76%. For species that are exclusively part of the validation set, the MARD is 9.93%

Table 1 Prediction performance of DeepESNet on different dataset splits. Data reported in percentage, calculated with eqn (11) and (12). Below the overall value, we added the results for species exclusively part of the validation or test set on the respective column

	MARD/ARD		
	Training	Validation	Test
Split 1	6.05%/4.02%	7.59%/4.20% 9.93%/4.55%	10.0%/4.76% 12.8%/5.66%
Split 2	5.78%/3.63%	7.71%/4.28% 8.75%/4.67%	7.81%/4.05% 8.74%/4.27%

and the median ARD is 4.55%. For species that are exclusively part of the test set, the MARD is 12.78% and the median ARD is 5.66%. The differences between train, test, and validation sets can typically be explained by a few outliers. The values are summarized on Table 1.

For the training data of Split 2, the MARD between model prediction and experimental data is 5.78% and the median ARD is 3.63%. For validation data, the MARD between model prediction and Experimental data is 7.71% and the median ARD is 4.28%. For test data, the MARD between model prediction and experimental data is 7.81% and the median ARD is 4.05%. For species that are exclusively part of the validation set, the MARD is 8.75% and the median ARD is 4.67%. For species that are exclusively part of the test set, the MARD is 8.74% and the median ARD is 4.27%. Predictions for selected species that are only part of the test data are shown in Fig. 6. The model and more detailed results can be found on GitHub.[59]

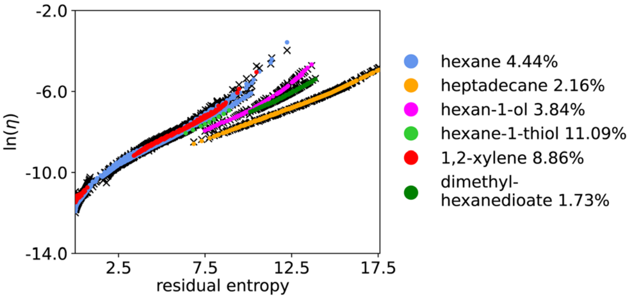

Fig. 6 Predictions for selected species that are only part of the test data to highlight the predictive capabilities of the model and the underlying architecture. MARD shown next to the substance name. For comparison, the MARD values reported for classical entropy scaling with group contribution method[26] are: 4.55% (hexane), 3.93% (heptadecane), 16.78% (hexan-1-ol).

A box plot showing the results for all available data grouped by species can be found in Fig. 7. The family with the highest median ARD is the naphtalenes, which are cyclic with several rings, *i.e.*, higher ARDs are expected. Other families with high median ARDs contain highly branched or cyclic molecules that are also not adequately represented by PC-SAFT parameters. On the other hand, simple cyclic molecules such as 1,2-xylene (see Fig. 6) and the components found in the family of other alkylbenzenes, show good agreement with experimental results. Species with atom types significantly underrepresented in the entire data set, for example iodine, were also found as outliers. This finding indicates that PC-SAFT parameters exhibit specific deficiencies in the representation of polycyclic and highly branched molecules.

7.3 Performance evaluation of DeepESNet architecture

7.3.1 Model comparison and dataset. The DeepESNet architecture was evaluated against different baseline approaches using 50/25(15)/25(15), 40/30(20)/30(20), and 20/40(25)/40(25) splits. All trained models and implementation code are publicly available.[59] The comparative analysis included three variants of the feed forward neural network, a modified DeepESNet architecture lacking communication between the two branch nets (from the reference GenDeepONet to the second GenDeepONet block of the DeepESNet architecture), and a DeepESNet architecture with trunk networks replaced by polynomials. All feed forward models utilized PC-SAFT

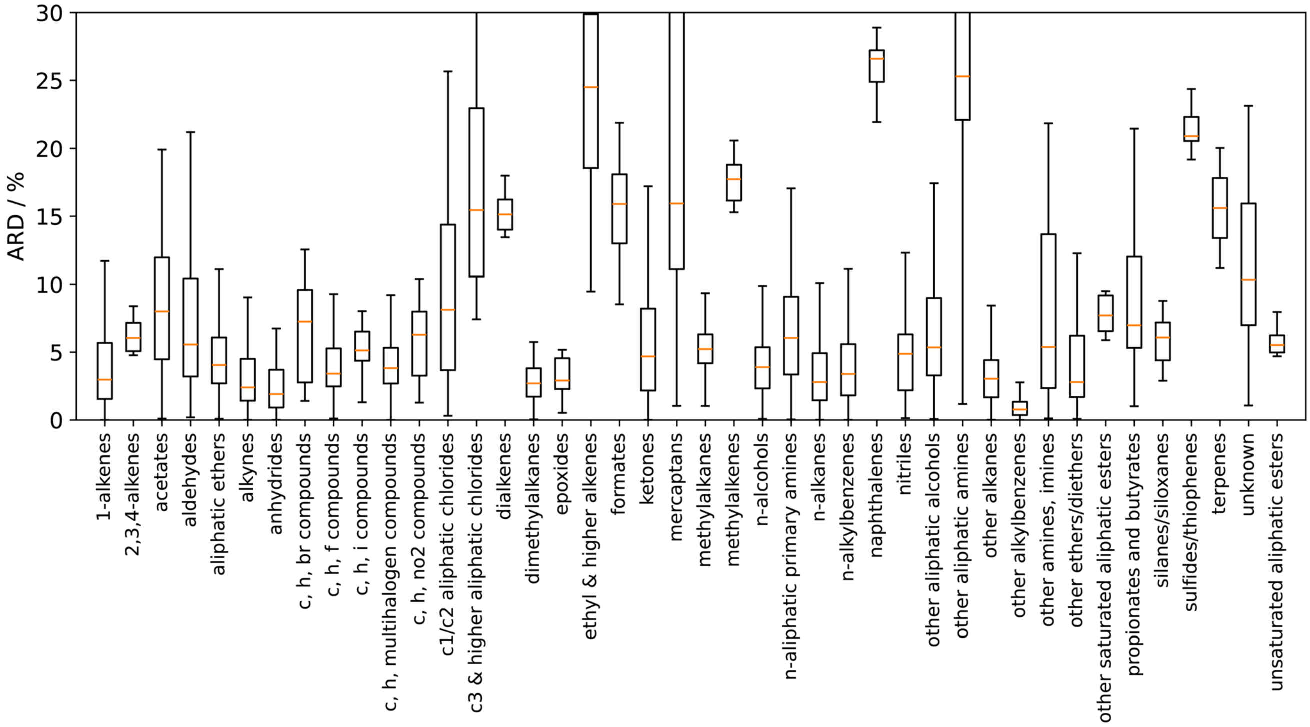

Fig. 7 Test, validation and training data grouped by families. Orange lines mark the median ARD of the respective family.

parameters, molar mass, and temperature as base inputs, with varying additional parameters: pressure, logarithmic pressure, or residual entropy. The comparative analysis demonstrated superior performance of the DeepESNet architecture across all evaluated metrics (SI Fig. S6–S19).

7.3.2 **Key findings.** The DeepESNet architecture showed training stability and better extrapolation performance for unknown species. This improvement is attributed to the sequential, directed, and meaningful communication between the networks of the multi-network framework. The modified DeepESNet without inter-branch communication showed intermediate performance, outperforming feed forward approaches while remaining less reliable than the fully connected architecture. Furthermore, replacing trunk networks with polynomials produces slimmer models – with fewer weights (13 078 on our study case) – that perform only slightly worse than the full DeepESNet architecture. The polynomials are of a very high order – for example, order 12 – which corresponds to the magnitude of the number of features used in the DeepESNet models. Although the high order is preferable, it can have a negative effect on extrapolation behavior.

Among feed forward models, performance ranked as follows: entropy-based > density-based > logarithmic pressure > pressure-based inputs, validating the efficacy of entropy scaling approaches even in simpler architectures.

7.3.3 **Data and computational efficiency**. Using a much smaller dataset to train the model, a 20/40(25)/40(25) split, *i.e.*, using only 20% of available samples, the DeepESNet maintained superior performance across all evaluation metrics. The degradation of performance with reduced training data was the smallest for DeepESNet net. While the metrics between DeepESNet and entropy feed forward are close for large training datasets, the differences become apparent with small training datasets.

The DeepESNet architecture also achieved very good performance with significantly reduced computational complexity. The optimal DeepESNet configuration (50/25(15)/25(15) split) used 18 286 parameters compared to 50 945 parameters in the best-performing entropy-based feed forward model. This represents a 64% reduction in model complexity.

7.3.4 **Comparison to preliminary work.** The preliminary architecture[31] was evaluated on smaller, more homogeneous datasets ($\approx$600 species, fewer chemical families). It incorporated feed forward corrections to the Chapman–Enskog reference, requiring full state-point information (temperature and density). This introduces overfitting risks while maintaining the limitations of feed forward approaches without enforcing entropy scaling principles.

The current work shows superior performance on a more challenging dataset and underwent a more thorough benchmark analysis, establishing the DeepESNet architecture as a significant advancement over the previous methodology and other feed forward alternatives.

7.4 Further utilization options

In process simulators, evaluating the viscosity of a list of substances at high frequency with low computational times is crucial. Consequently, parameters obtained from the branch networks only require initialization at the start of a simulation, whereas during the simulation only the small trunk nets need to be called. As the inputs of trunk networks are one-dimensional, they can be replaced by polynomials for additional speed-ups. In both cases, either the full DeepESNet architecture or the denoising architecture plus a polynomial fitted to the denoised viscosity (η^*) can be used as model.

The models trained in this work rely on PC-SAFT parameters obtained from SMILES strings with transformers from Winter *et al.*[53] As the model inputs are PC-SAFT parameters, the model can be employed as a quasi-equation of state for viscosities or retrained with alternative PC-SAFT parameter sets and different experimental data. If alternative PC-SAFT parameter sets are used, it is necessary to retrain the model with the respective data, as the PC-SAFT parameters vary from set to set. We also would like to emphasize that the PC-SAFT parameters from Winter *et al.* are machine-generated and are therefore more suitable for machine learning than sets based on individual substance fits.

It is important to note that all of the possible uses illustrated in this paper can be combined as desired in order to obtain ideal models for the respective application.

8 Conclusions

In this work, a generalized entropy scaling (DeepESNet) architecture for viscosities was developed, implemented, and tested. The generalized model consists of two linked generalized DeepONets (GenDeepONet) architectures, with enhanced predictive capabilities and simplified training compared to feed forward neural networks. The DeepESNet architecture showed training stability and better extrapolation performance for unknown species compared to classical analytical entropy scaling. In addition to the capacity for predicting the viscosities of other substances, the model can also be utilized to obtain highly accurate models of well-measured substances and substance families, similarly to a substance-specific fit in analytical entropy scaling. Furthermore, simplified architectures can be trained that are highly efficient in terms of computing time. The wide range of potential applications of this approach underscores its versatility and flexibility, which are direct consequences of the generalization of the entropy scaling principle. This approach offers significant potential to address a broad spectrum of problems and applications.

Author contributions

M. F.: conceptualization, methodology, software, investigation, formal analysis, writing – original Draft; M. S.: visualization, writing – review & editing; S. D.: methodology, software; T. K.: methodology; N. H.: supervision, writing – review & editing, funding acquisition.

Conflicts of interest

There are no conflicts to declare.

Data availability

Data for this article, including full implementation of the models discussed in this work, are available at GitHub at https://github.com/maxfleck/deep-entropy-scaling.git (DOI: https://doi.org/10.5281/zenodo.17419467).

Supplementary information: Further details on the ML model, performance comparison, outlier detection, and previous work. See DOI: https://doi.org/10.1039/d5dd00179j.

Acknowledgements

This work was funded by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2075 – 390740016. We acknowledge the support by the Stuttgart Center for Simulation Science (SimTech) and 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 DFG through grant no INST 37/935-1 FUGG.

References

1 J. M. Prausnitz, *Angew. Chem., Int. Ed.*, 1990, **29**, 1246–1255.

2 H.-C. Hsu, Y.-W. Sheu and C.-H. Tu, *Chem. Eng. J.*, 2002, **88**, 27–35.

3 V. R. Bhethanabotha, *A group contribution method for liquid viscosity*, Pennsylvania State University Press, 1983.

4 D. Van Velzen, R. L. Cardozo and H. Langenkamp, *Ind. Eng. Chem. Fundam.*, 1972, **11**, 20–25.

5 L. H. Thomas, *J. Chem. Soc.*, 1946, 573–579.

6 S. Sastri and K. Rao, *Chem. Eng. J.*, 1992, **50**, 9–25.

7 S. Sastri and K. Rao, *Fluid Phase Equilib.*, 2000, **175**, 311–323.

8 J. Przedziecki and T. Sridhar, *AIChE J.*, 1985, **31**, 333–335.

9 A. Teja and P. Rice, *Ind. Eng. Chem. Fundam.*, 1981, **20**, 77–81.

10 M.-J. Lee and M.-C. Wei, *J. Chem. Eng. Jpn.*, 1993, **26**, 159–165.

11 D. J. Carlson, N. F. Giles, W. V. Wilding and T. A. Knotts IV, *Fluid Phase Equilib.*, 2022, **561**, 113522.

12 D. J. Carlson, N. F. Giles, W. V. Wilding and T. A. Knotts IV, *Fluid Phase Equilib.*, 2023, **566**, 113681.

13 M. Fischer, G. Bauer and J. Gross, *Ind. Eng. Chem. Res.*, 2020, **59**, 8855–8869.

14 S. Schmitt, F. Fleckenstein, H. Hasse and S. Stephan, *J. Phys. Chem. B*, 2023, **127**, 1789–1802.

15 M. B. Spera, S. Darouich, J. Pleiss and N. Hansen, *Fluid Phase Equilib.*, 2025, **592**, 114324.

16 M. Fleck, S. Darouich, J. Pleiss, N. Hansen and M. B. Spera, *J. Chem. Inf. Model.*, 2025, **65**, 3999–4009.

17 A. M. Schweidtmann, E. Esche, A. Fischer, M. Kloft, J.-U. Repke, S. Sager and A. Mitsos, *Chem. Ing. Tech.*, 2021, **93**, 2029–2039.

18 A. K. Chew, M. Sender, Z. Kaplan, A. Chandrasekaran, J. Chief Elk, A. R. Browning, H. S. Kwak, M. D. Halls and M. A. F. Afzal, *J. Cheminf.*, 2024, **16**, 31.

19 Y. Rosenfeld, *Phys. Rev. A*, 1977, **15**, 2545.

20 Y. Rosenfeld, *J. Phys.: Condens. Matter*, 1999, **11**, 5415.

21 M. Hopp, J. Mele and J. Gross, *Ind. Eng. Chem. Res.*, 2018, **57**, 12942–12950.

22 M. Hopp and J. Gross, *Ind. Eng. Chem. Res.*, 2019, **58**, 20441–20449.

23 O. Lötgering-Lin, M. Fischer, M. Hopp and J. Gross, *Ind. Eng. Chem. Res.*, 2018, **57**, 4095–4114.

24 S. Schmitt, H. Hasse and S. Stephan, *Nat. Commun.*, 2025, **16**, 2611.

25 M. Fleck, J. Gross and N. Hansen, *Ind. Eng. Chem. Res.*, 2024, **63**, 3755–3765.

26 O. Lötgering-Lin and J. Gross, *Ind. Eng. Chem. Res.*, 2015, **54**, 7942–7952.

27 I. H. Bell, *J. Chem. Eng. Data*, 2020, **65**, 3203–3215.

28 I. H. Bell, *J. Chem. Eng. Data*, 2020, **65**, 5606–5616.

29 I. H. Bell, R. Fingerhut, J. Vrabec and L. Costigliola, *J. Chem. Phys.*, 2022, **157**, 074501.

30 A. Dehlouz, R. Privat, G. Galliero, M. Bonnissel and J.-N. Jaubert, *Ind. Eng. Chem. Res.*, 2021, **60**, 12719–12739.

31 M. Fleck, J. Gross and N. Hansen, *ChemRxiv*, 2024, DOI: 10.26434/chemrxiv-2024-8982t.

32 S. Schmitt, H. Hasse and S. Stephan, *J. Mol. Liq.*, 2024, **395**, 123811.

33 J. Gross and G. Sadowski, *Ind. Eng. Chem. Res.*, 2001, **40**, 1244–1260.

34 D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik and R. P. Adams, *Advances in Neural Information Processing Systems*, 2015.

35 M. Krenn, F. Häse, A. Nigam, P. Friederich and A. Aspuru-Guzik, *Mach. Learn.: Sci. Technol.*, 2020, **1**, 045024.

36 W. A. Malatesta and B. Yang, *ACS Omega*, 2021, **6**, 28579–28586.

37 L. Novak, *Int. J. Chem. React. Eng.*, 2011, **9**, A63.

38 J.-L. Bretonnet, *J. Chem. Phys.*, 2002, **117**, 9370–9373.

39 R. Chopra, T. M. Truskett and J. R. Errington, *J. Phys. Chem. B*, 2010, **114**, 10558–10566.

40 R. Chopra, T. M. Truskett and J. R. Errington, *Phys. Rev. E*, 2010, **82**, 041201.

41 T. Goel, C. N. Patra, T. Mukherjee and C. Chakravarty, *J. Chem. Phys.*, 2008, **129**, 164904.

42 W. P. Krekelberg, T. Kumar, J. Mittal, J. R. Errington and T. M. Truskett, *Phys. Rev. E*, 2009, **79**, 031203.

43 W. P. Krekelberg, M. J. Pond, G. Goel, V. K. Shen, J. R. Errington and T. M. Truskett, *Phys. Rev. E*, 2009, **80**, 061205.

44 M. J. Pond, J. R. Errington and T. M. Truskett, *J. Chem. Phys.*, 2011, **134**, 081101.

45 S. Pieprzyk, D. Heyes and A. Brańka, *Phys. Rev. E*, 2014, **90**, 012106.

46 M. Hopp and J. Gross, *Ind. Eng. Chem. Res.*, 2017, **56**, 4527–4538.

47 J. M. Young, I. H. Bell and A. H. Harvey, *J. Chem. Phys.*, 2023, **158**, 024502.

48 J. O. Hirschfelder, C. F. Curtiss and R. B. Bird, *Molecular Theory of Gases and Liquids*, Wiley, New York, 1964.

49 S. Chapman and T. G. Cowling, *The Mathematical Theory of Non-Uniform Gases: an Account of the Kinetic Theory of*

Viscosity, Thermal Conduction and Diffusion in Gases, Cambridge University Press, 1990.

50 K. Fukushima, *Biol. Cybern.*, 1975, **20**, 121–136.

51 S. Wang, H. Wang and P. Perdikaris, *Sci. Adv.*, 2021, **7**, eabi8605.

52 M. Zhu, H. Zhang, A. Jiao, G. E. Karniadakis and L. Lu, *Comput. Methods Appl. Mech. Eng.*, 2023, **412**, 116064.

53 B. Winter, P. Rehner, T. Esper, J. Schilling and A. Bardow, *Digital Discovery*, 2025, **4**, 1142–1157.

54 DDB, Dortmund Data Bank, 2022, **http://www.ddbst.com**.

55 M. Frenkel, R. D. Chiroco, V. Diky, Q. Dong, K. N. Marsh, J. H. Dymond, W. A. Wakeham, S. E. Stein, E. Königsberger and A. R. Goodwin, *Pure Appl. Chem.*, 2006, **78**, 541–612.

56 G. Thomson, *Int. J. Thermophys.*, 1996, **17**, 223–232.

57 D. P. Kingma and J. Ba, *arXiv*, 2014, preprint arXiv:1412.6980, DOI: **10.48550/arXiv.1412.6980**.

58 A. Y. Ng, *Proceedings of the twenty-first international conference on Machine learning*, 2004, p. 78.

59 M. Fleck, **https://github.com/maxfleck/deep-entropy-scaling.git**, accessed on 08/07/2025.