Application of machine-learning algorithms to predict the transport properties of Mie fluids

Justinas Šlepavičius,[1] Alessandro Patti,[1,2] James L. McDonagh,[3] and Carlos Avendaño[1, a)]

[1)]*Department of Chemical Engineering, School of Engineering, The University of Manchester, Oxford Road, Manchester, M13 9PL, United Kingdom*

[2)]*Department of Applied Physics, University of Granada, Fuente Nueva s/n, 18071 Granada, Spain*

[3)]*IBM Research Europe, The Hartree Centre STFC Laboratory Sci-Tech Daresbury, Warrington, United Kingdom* [b)]

(Dated: 9 June 2023)

The ability to predict transport properties of fluids, such as the self-diffusion coefficient and viscosity, has been an ongoing effort in the field of molecular modelling. While there are theoretical approaches to predict the transport properties of simple systems, they are typically applied in the dilute gas regime and are not directly applicable to more complex systems. Other attempts to predict transport properties are done by fitting available experimental or molecular simulation data to empirical or semi-empirical correlations. Recently, there have been attempts to improve the accuracy of these fittings through the use of Machine Learning (ML) methods. In this work, the application of ML algorithms to represent the transport properties of systems comprising spherical particles interacting *via* the Mie potential is investigated. To this end, the self-diffusion coefficient and shear viscosity of 54 potentials are obtained at different regions of the fluid-phase diagram. This data set is used together with three ML algorithms, namely *k*-Nearest Neighbours, Artificial Neural Network and Symbolic Regression, to find correlations between the parameters of each potential and the transport properties at different densities and temperatures. It is shown that ANN and KNN perform to a similar extent, followed by SR, which exhibits larger deviations. Finally, the application of the three ML models to predict the self-diffusion coefficient of small molecular systems, such as krypton, methane and carbon dioxide is demonstrated using molecular parameters derived from the so-called SAFT-VR Mie equation of state [J. Chem. Phys. **139**, 154504 (2013)] and available experimental vapour-liquid coexistence data.

[a)]Electronic mail: carlos.avendano@manchester.ac.uk

[b)]Current address: Ladder Therapeutics doing business as Serna Bio, Lab F37, Stevenage Bioscience Catalyst, Gunnels Wood Road, Stevenage, Hertfordshire, SG1 2FX, United Kingdom

I. INTRODUCTION

Accurate representation of thermodynamic and transport properties of molecular systems is key for the design and optimization of chemical and biochemical processes and plays an important role in several areas of science and technology[1,2]. For many decades, most attention has been drawn to the development of robust thermodynamic models of fluids based on activity coefficient models[3], cubic equations of state[4,5], and molecular-based models[6–9]. The development of models for the description of transport properties has also been reported in recent years, particularly for high-density liquids. Many of these approaches are based on kinetic theory, mode-coupling theory, free-volume theory, and friction theory[10–13].

Molecular dynamics is a powerful technique to determine the transport properties of molecular systems using either Green-Kubo or Einstein relations[14,15], provided a suitable force field is available. The simulations can be used to test theoretical predictions of transport properties. However, molecular simulations can be computationally expensive and not suitable for fast property prediction required in many industrial applications. For this reason, simple empirical and semi-empirical correlations are still commonly used for many engineering calculations, but these equations have a limited range of applicability making them unreliable for the design of novel technologies.

There have been previous attempts to obtain theoretical or semi-empirical expressions for the self-diffusion coefficients of simple systems, such as hard spheres, Lennard-Jones, WCA, and molecular and atomic species[16–27]. Still, large amounts of data are required for fitting the expressions that typically have complicated functional forms with multiple fitting parameters. A similar situation has been observed in the developments of semi-empirical equations for the viscosity[28–31], however, new advances in the use of equations of state coupled with methods such as the free-volume, kinetic, and friction theories to predict shear viscosity of complex molecular systems have been reported, particularly for the prediction of the shear viscosity[32–34].

To facilitate the way the predictions about the behaviour of transport properties are made, attention has been turning to machine learning (ML) methods. These methods are able to take the large amounts of data available and explore possible correlations between the system parameters and properties of interest. This change towards ML methods has already been taking place in similar fields, such as in the prediction of physicochemical properties[35–54]. For a recent account on this topic, the reader is directed to the recent reviews by Schmidt *et al.*[55], Moud *et al.*[56] and Ihme *et al.*[57] The use of ML to predict transport properties, such as the self-diffusion coefficient,

has been studied to a much lesser extent[58]. ML algorithms to study transport properties have been used to predict diffusion in binary hydrocarbons[59], water mixtures[60], polar and non-polar binary gases[61], organic molecules[62,63], and CO_2[64]. Recently, the application of ML algorithms to study the self-diffusion coefficient of Lennard-Jones systems has been reported[65–68], where the accuracy of the predictions was increased by more than one order of magnitude compared to semi-empirical equations.

In this work, ML methods are applied to predict the transport properties, particularly self-diffusion coefficient and viscosity, of spherical particles interacting *via* the Mie potential[69]. The Mie potential is a generalized form of the Lennard-Jones potential but with the advantage that the repulsive and attractive contributions can be tuned to represent accurately the properties of real molecular systems. This potential has been used to study successfully the behaviour of small molecular systems (H_2O, CO_2, SF_6, and CF_4)[70–73], and is also the underlying intermolecular potential used in several force fields and molecular-based theories for the description of properties of complex molecular systems[9,74,75]. The different combinations of the repulsive and attractive contributions impart a large variability of the potential, which also means that there is not a large amount of published data, especially of transport properties, that can be used in this approach using ML methods. Only a small amount of published research in the literature report data on many different types of Mie potentials since most reports concentrate on modelling real particles using a specific combination of potential parameters, hence only a fraction of the potential space is explored and published in the literature. Therefore, in this work, the data required to fit the ML models are also determined.

This paper is organised as follows. In Sec. II, the simulation methods employed to obtain the self-diffusion coefficient and shear viscosity data required for the ML algorithms are described. In the same section, the different ML algorithms applied in this work are discussed. In Sec. III, the performance of the ML algorithms in each of the cases studied is reported, and the application of the methods to describe the self-diffusion coefficient of three quasi-spherical systems, namely krypton Kr, methane CH_4, and carbon dioxide CO_2 is discussed. Finally, in Sec. IV, the conclusions of the work are presented.

II. METHODS

A. Molecular-dynamics simulations

The molecular model consists of spherical particles interacting via the Mie potential given by[69]

$$\phi(r) = \mathscr{C}\varepsilon\left[\left(\frac{\sigma}{r}\right)^{n} - \left(\frac{\sigma}{r}\right)^{m}\right], \tag{1}$$

where r is the interparticle distance, ε is the potential well-depth, σ is the diameter of the particles, and the exponents n and m describe the range of the repulsive and attractive contributions of the potential, respectively. The coefficient $\mathscr{C}$ in Equation 1 is given by

$$\mathscr{C} \equiv \left(\frac{n}{n-m}\right)\left(\frac{n}{m}\right)^{\frac{m}{n-m}}, \tag{2}$$

and is defined such that the minimum of the potential is set at $-\varepsilon$. The Mie(n,m) potential is a generalised form of the well-known Lennard-Jones (LJ) potential, which is obtained when $n = 12$ and $m = 6$. It has been demonstrated that the use of different repulsive and attractive exponents results in a much better description of the intermolecular interactions of complex systems[9,70,71,75]. While there is freedom in selecting arbitrary values of (n,m) to represent a particular molecular system, it has been shown that there are different combinations of exponents that can lead to the same critical point, albeit different triple points[76]. In particular, Ramrattan *et al.* showed that combinations of exponents with the same cohesive parameter α, which is defined as

$$\alpha \equiv \mathscr{C}\left(\frac{1}{m-3} - \frac{1}{n-3}\right), \tag{3}$$

exhibit identical critical points[76]. In other words, fluids with the same cohesive parameter are conformal. Due to this observation, it is hypothesised that pair of exponents (n,m) with the same cohesive parameter α should also exhibit the same transport properties in the fluid region akin to the principle of corresponding states for transport properties[77,78]. To determine the critical point and fluid phase diagram associated with a particular combination of exponents (n,m), the SAFT-VR Mie equation of state (EoS) is employed[9]. This EoS has been shown to represent the critical region accurately due to the high-order terms considered in the Barker and Henderson perturbation theory[79].

Throughout this work, reduced units are employed to describe thermodynamic and transport properties using the Mie potential parameters: number density $\rho^* = N\sigma^3/V$, temperature $T^* = k_{\mathrm{B}}T/\varepsilon$, pressure $p^* = p\sigma^3/\varepsilon$, time $t^* = t[\varepsilon/(m\sigma^2)]^{1/2}$, self-diffusion coefficient

$D^* = D[m/(\sigma^2\varepsilon)]^{1/2}$, and viscosity $\eta^* = \eta\sigma^2/(\varepsilon m)^{1/2}$, where N is the total number of particles, V is the volume of the system, T is the absolute temperature, k_B is the Boltzmann constant, p is the absolute pressure, t is the time, m is the mass of a spherical particle, D is the self-diffusion coefficient, and η is the shear viscosity. Similarly, all distances are given in units of σ. To determine the transport properties of the Mie potential, molecular-dynamics simulations are performed in systems comprising $N = 10^4$ particles in the canonical NVT ensemble using the Nosé-Hoover thermostat[80]. For the simulations, the Mie potential is truncated and shifted to zero using a cut-off of $r_c^* = 6$. All the simulations are performed using the LAMMPS package[81]. The equations of motion are integrated using the velocity-Verlet algorithm with a time step of $\Delta t = 0.001\tau$. For each state point, five independent simulations of 10^7 time steps are performed to collect averages.

The self-diffusion coefficient has been calculated using the Einstein equation given by

$$D^* = \frac{1}{6t^*}\langle \Delta r^{*2}(t^*)\rangle, \tag{4}$$

where $\langle \Delta r^{*2}(t^*)\rangle$ is the ensemble average of the mean-square displacement (MSD) given by

$$\langle \Delta r^{*2}(t^*)\rangle = \frac{1}{N}\Big\langle \sum_{j=1}^{N} [\mathbf{r}_j^*(t^*) - \mathbf{r}_j^*(0)]^2 \Big\rangle, \tag{5}$$

and $\mathbf{r}_j^*(t^*)$ is the position of particle j at time t^*. The trajectories used to calculate the MSD are considered independent when the system enters the diffusive regime from the ballistic, i.e., when $\langle \Delta r^{*2}\rangle \propto t^{*2}$ changes to $\langle \Delta r^{*2}\rangle \propto t^*$. In this systems, the diffusive regime starts at $t^* \approx 1$, meaning at least 1000 timesteps are required to reach the change in regime. To ensure our trajectories are uncorrelated, new simulations are started after 10^5 timesteps have elapsed, which is 100 times longer than the average time to decorrelation.

The shear viscosity is calculated through the Green-Kubo relation of the time-correlation of the off-diagonal elements of the pressure tensor given by[15,82,83]

$$\eta^* = \frac{V^*}{T^*}\int_0^\infty \langle p_{\alpha\beta}^*(t^*) p_{\alpha\beta}^*(t_0^*)\rangle \mathrm{d}t^* \tag{6}$$

where $p_{\alpha\beta}^*(t^*)$, with $\alpha \neq \beta$, is the off-diagonal component of the pressure tensor at time t^*.

The training data for the ML models have been collected from 54 different Mie(n,m) potentials. For each potential, 9 state points are sampled in the supercritical region, 5 state points in the liquid phase, and also 5 states in the vapour phase. The state points for the supercritical phase are taken at temperatures of T^*/T_c^*=$\{1.05, 1.25, 1.5\}$ and densities of ρ^*/ρ_c^*=$\{0.2, 1, 2\}$ for each potential, where T_c and ρ_c are the critical temperature and critical density obtained using the SAFT-VR Mie

EoS[9], respectively. For liquid and vapour phases, 5 state points per potential are chosen at random to ensure that there is no bias in the data collection. For the liquid phase, the state points are chosen by taking random points in the liquid side of the VLE, in the range of $T^*/T_c^* = 0.8-0.95$ and $\rho^*/\rho_c^* = 1.7-2.5$, while ensuring that the resulting points are at least 5% away from the saturated liquid density. A similar approach is employed to study the vapour phase using the same temperature range ($T^*/T_c^* = 0.8-0.95$) for densities $\rho^*/\rho_c^* < 0.7$, and ensuring that the selected density is $\rho^* > 0.005$ and at least 5% away from the saturated vapour density. Note that the shear viscosity is only computed and analyzed for the liquid-state region. The following combinations of Mie exponents are studied: $\{n = 12-14, m = 6-8\}$, $\{n = 15-17, m = 6-11\}$ and $\{n = 18-20, m = 6-14\}$. These combinations ensure that at least $n-m \geq 4$ and also cover enough parameter space for the study of real molecular systems. All the transport properties of the Mie potentials determined in this work as well as the ML model files are reported in the Supplementary Information (SI) and are available on GitHub.

B. Machine Learning

In this work, three algorithms to predict the transport properties of Mie fluids are assessed: *k*-nearest neighbours (KNN), artificial neural network (ANN), and symbolic regression (SR). These algorithms are chosen due to their different levels of complexity and interpretability. KNN and ANN are both "black box" methods, which makes them very difficult to interpret, without studying the resulting algorithm in depth, whereas SR provides a straightforward correlation equation. For all the algorithms, 80% of the data are used to train the model and 20% to test the model. For the case of ANN and KNN, the data is normalized to be in the interval [-1,1]. Before training any model, the self-diffusion coefficient and the shear viscosity are normalized using Chapman-Enskog expressions for the self-diffusion coefficient D_0^* and the shear viscosity η_0^* of a dilute gas of hard spheres, respectively, given by[10,20]

$$D_0^* = \frac{3}{8}\left(\frac{T^*}{\pi}\right)^{1/2}\frac{1}{\rho^*} \tag{7}$$

and

$$\eta_0^* = \frac{5}{16}\left(\frac{T^*}{\pi}\right)^{1/2} \tag{8}$$

In other words, all ML methods explored in this work are trained in D/D_0 for the self-diffusion, and in η/η_0 for the viscosity. This semi-empirical approach has also been used by other authors to

study systems such as hard spheres[19] and Lennard-Jones fluids[84]. The quantification of errors and accuracy of the models, however, are reported with respect to D^* and η^*. The heat maps of the errors observed using the three different ML in the phase space are presented in the Supplementary Information, as well as the prediction of our method to predict available transport properties of LJ particles.

The performance of the ML algorithms with respect to the testing data has been quantified using the coefficient of determination R^2 as well as the absolute average relative deviation (AARD) defined as

$$\text{AARD} = \frac{1}{n}\sum_{i=1}^{n}\frac{|(y_i - \hat{y}_i)|}{y_i} \times 100\%, \tag{9}$$

where n is the number of samples, and y_i and $\hat{y}_i$ indicate the true and predicted values of the sample i, respectively. Both AARD and R^2 are used in assessing the performance of ML algorithms and, in the case of ANN and KNN, are also used in determining the accuracy of a learning method in itself. The calculation of method-training accuracy is done using 10-fold cross-validation (CV10), which entails dividing the test data into 10 different randomly selected and equal-sized sections, training the model with 9 of the sections and validating with the 10^{th} section. This process is repeated until the algorithm uses all sections as validation data, thus allowing for the calculation of the performance using AARD. Both ANN and KNN methods have been implemented in Python 3 using the *scikit-learn* library version 1.2.2[85].

The complexity and performance of the ANN are largely influenced by the number of hidden layers and the number of neurons each layer has. Additional layers provide the ability for the ANN to capture more complex input/output dependencies but require more time and data to train. Additionally, more complex ANN architectures may lead to data over-fitting if the underlying correlations are less complex, so the choice of architecture needs to be optimized. In this work, the performances of networks with different numbers of hidden layers and numbers of neurons in the hidden layers are quantified using AARD as the metric for comparison. It is found that a neural network with a single hidden layer consisting of 28 nodes is sufficient and increasing the complexity does not significantly improve the performance. The activation function used is the reLU function, the training is done for 1000 epochs and the *lbfgs* solver is used for back-propagation.

The KNN algorithm can be used for both classification and regression, and uses a number of k nearest neighbours to perform interpolation to predict a new state. In classification, the method

uses the distances to these k neighbours to classify the test point, while in regression problems, the values of the neighbours are weighted by the distances to the neighbours to predict the value of the test point. The hyperparameters of the KNN algorithm play a primary role in the performance and efficiency of the algorithm. To select the most appropriate hyperparameters, CV10 is used to find the combination of hyperparameters that lead to the lowest AARD. The hyperparameters validated are the value of k, the weights assigned to the data points, the algorithm to find the closest points, and the power parameter for the Minkowski metric. During the validation, none of the hyperparameters tested shows a significant drop in performance in both quality and speed, with the largest difference in performance only observed when changing the value of k. The hyperparameters that provide the best performance are k= 4, the neighbouring points weighted by distance, and the power parameter of 4 in the Minkowski metric. The algorithm used for the closest search point does not make any appreciable difference as long as the brute-force search is not used.

SR is a very different ML algorithm compared to KNN and ANN. Rather than learning the connections between outputs and inputs in the traditional ML sense, it attempts to find a mathematical expression that correlates the behaviour of the output given the inputs by building a binary tree, in which leaves are constants or inputs and branches are mathematical operations. To find the most appropriate equation, the method starts with naive random guesses and uses evolution and mutations to obtain more accurate equations. In this work, the SR implementation from the *gplearn* library version 0.4.1 is used[86]. The hyperparameter space for SR is very large, as the mutation and evolution steps can be changed to allow for more or less variability between different members of the population and between parents and offspring equations. Other hyperparameters control the population size, the number of generations of evolution, the types of functions that can be used to connect the nodes, the parsimony coefficient, which is a measure of the 'complexity' of the final equation, and the metric by which the performance of each member of the population is evaluated. Each hyperparameter needed to be checked individually not only for accuracy but also for variability in obtained equations in different random states and the length of the evolution. After some initial testing, it is found that a population size of 5000, 50 generations, the AARD metric and a parsimony coefficient of 0.3 are the hyperparameters that affect the algorithm the most. The operation set used in this work comprises additions, subtractions, multiplications, divisions, exponential, square roots of the absolute values and the natural logarithm of the absolute values.

III. RESULTS

A. Self-diffusion coefficient

1. Selection of relevant features for the ML methods and conformality of the Mie potential

The correct selection of features is critical in ML. For fluids comprising spherical particles interacting *via* the Mie potential, the obvious features that can be used are temperature T^* and density ρ^* to define the thermodynamic state, and both repulsive n and attractive m exponents to define the intermolecular potential. Neither σ nor ε is needed since all the properties are expressed in reduced units. However, as discussed in Section II A, it has been demonstrated that Mie potentials with different pairs of exponents (n,m) that lead to the same value of the cohesive parameter α (Equation 2) are conformal and exhibit the same critical points and, potentially, should lead to the same transport properties in the fluid phase. To corroborate this hypothesis, the self-diffusion coefficients obtained from the MD simulations of three triplets of Mie potentials are studied. Here, each triplet has the same value of the cohesive parameter α corresponding to $\alpha = 0.899$, 0.585, and 0.254 but within each set, the repulsive and attractive exponents differ considerably from one another. These sets of potentials have been studied at a subcritical isotherm $(T^*/T_c^* = 0.95)$ and at a supercritical isotherm $(T^*/T_c^* = 1.2)$ for a wide range of densities, and the results are presented in Figure 1. Despite the small deviations observed at very high densities near the freezing points, the agreement of the self-diffusion coefficient within a set of potentials with an identical value of the cohesive parameter is remarkable, particularly for densities corresponding to $\rho^* < 0.7$, which is to the region in which most of the simulation data has been collected.

The predictive power of the cohesive parameter α to describe the self-diffusion coefficient of different potentials reaffirms the need to use it as a feature in the ML algorithm training rather than using the individual exponents n and m. The relation between the cohesive parameter α and the exponents is non-linear, as shown in Equation 3, hence using the individual exponents introduces a layer of complexity between inputs and outputs, that may result in a lower ability of ML algorithms to predict transport properties.

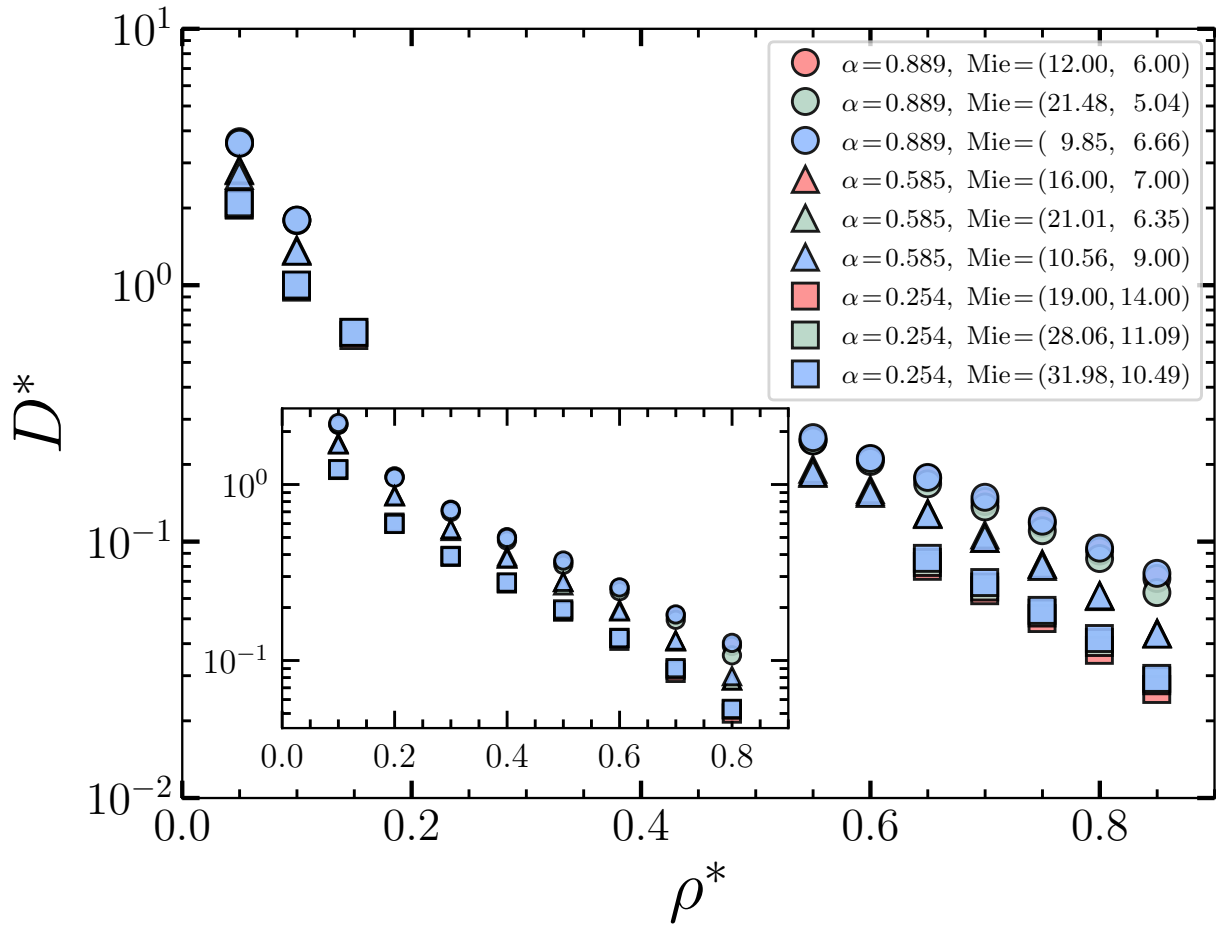

FIG. 1. MD simulation results of the self-diffusion coefficient D^* as a function of the density ρ^* obtained for three triplets of potentials with different cohesive parameter α. The results for the subcritical isotherm $T^*/T_c^* = 0.95$ are presented in the main figure, while the results for the supercritical isotherm $T^*/T_c^* = 1.2$ are shown in the inset.

2. *k-Nearest Neighbours and Artificial Neural Networks*

The training of the ANN and KNN algorithms using the MD results for the self-diffusion coefficient has been performed first according to three regions in the phase diagram, namely the subcritical vapour phase, the subcritical liquid phase and the supercritical fluid phase, and then analysed the entire set as a whole. As explained in the previous section, the features used in both algorithms are the temperature T^*, the density ρ^*, and the cohesive parameter α and the output is the Chapman-Eskog regularised self-diffusion coefficient (D^*/D_0^*).

First, the application of the ANN and KNN algorithms on the vapour phase data set is discussed.

TABLE I. Summary of the AARD and R^2 descriptors of different ML methods applied in this work.

Property	State	KNN		ANN		SR	
		AARD	R^2	AARD	R^2	AARD	R^2
D^*	Vapour phase	0.42%	0.9998	0.87%	0.9998	3.5%	0.995
D^*	Liquid phase	3.3%	0.988	1.8%	0.998	3.5%	0.992
D^*	Supercritical phase	3.4%	0.989	8.4%	0.977	7.7%	0.970
D^*	All phases	2.8%	0.9997	6.4%	0.998	30%	0.995
η^*	Liquid phase	2.8%	0.977	3.3%	0.988	6.3%	0.964

The results are presented in Figure 2(a) in the form of parity plots. It is evident from this Figure that in the vapour phase the ANN algorithm performs very similarly to KNN as seen from the summary of the AARD and R^2 presented in Table I. The AARD and R^2 for the ANN are 0.87% and 0.9998 for the vapour phase, respectively, compared to 0.42% and 0.9998 obtained using KNN. The algorithms differ in the accuracy of the prediction, to which AARD is more sensitive. The fact that KNN has a lower AARD may be explained by the fact that KNN uses interpolation to learn points to predict new points, rather than predicting the values based on inputs. The differences are minor, however, showing a good ability of both methods to predict the self-diffusion coefficient.

For the liquid phase, the algorithms do not predict D^* as well as for the vapour phase, with all metrics decreasing for both methods. The results are presented in Figure 2(b). The decrease in accuracy of the methods is contributed by an increased variation of the range of D^*/D_0^* in the liquid phase. Since the magnitude of D^* in the liquid phase is about two orders of magnitude smaller than in the vapour phase, the values of AARD are larger due to the amplification of the errors. It is observed that R^2 also decreases for the liquid phase, which is likely due to the normalization of D^*/D_0^* between [-1,1] where any deviations in the prediction of the methods will be amplified due to the increased range of the data used ($D^*/D_0^* = [0.08 - 0.56]$ for liquid phase compared to $D^*/D_0^* = [0.59 - 0.77]$ for the vapour phase). Additionally, it is observed that the decrease in accuracy is larger for KNN, particularly in the region of of low values of D^* and around $D^* \approx 0.1$ where data points are further away from the main diagonal in the parity plots.

The MD simulation data for the supercritical fluid phase has been collected at specific values of T^*/T_c^* and ρ^*/ρ_c^*. While the value of T_c^* varies greatly with the cohesive parameter α, the critical density remains mostly constant[87]. Therefore, the ranges of the self-diffusion coefficient

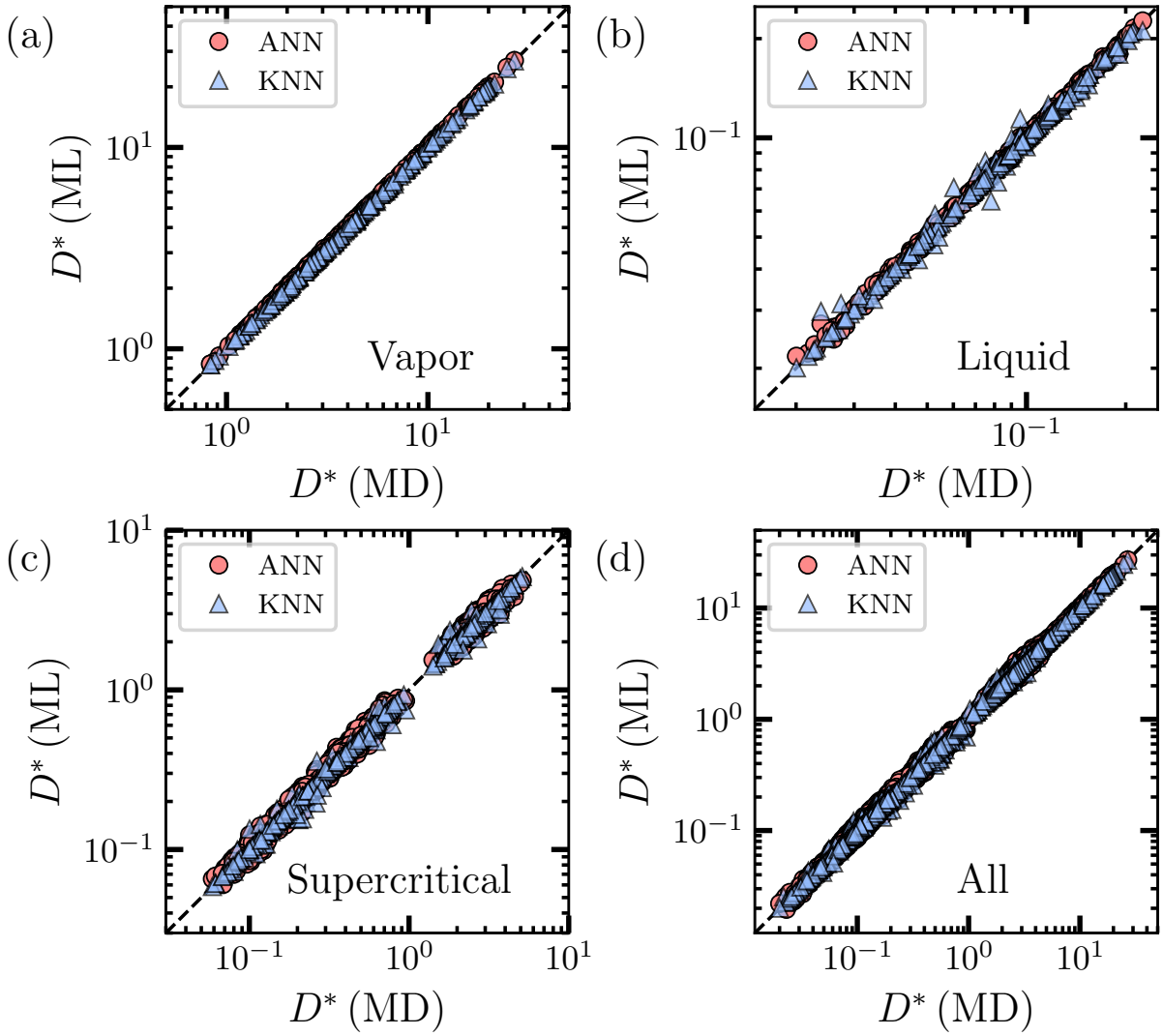

FIG. 2. Parity plots describing the performance of the ANN (red circles) and KNN (blue triangles) on the testing set of the self-diffusion coefficient for three different regions in the phase diagram, as well as for the entire data set as a whole. The results correspond to (a) subcritical vapour, (b) subcritical liquid, (c) supercritical fluid, and (d) all states. D^* (MD) denotes the self-diffusion coefficients obtained from MD simulations, while D^* (ML) denotes the predictions using machine learning.

are much more well-defined. The performance of ANN decreases for the supercritical phase with respect to the liquid phase, while KNN performs similarly, as shown Table I. It is worth noting that the performance of the methods fitting D^*/D_0^* is slightly lower than when trained on D^* alone only for the supercritical phase (data not shown).

Finally, the application of both ANN and KNN algorithms to describe the entire data set in

the three regions of interest is discussed. Taking the entire data set vastly increases the amount of training data available to the algorithms, while increasing the range of output values as well. The performance of the methods is observed in the parity plot in Figure 2(d), where it can be observed that both KNN and ANN are able to similarly predict the value of D^* with KNN slightly outperforming the ANN algorithm.

3. Symbolic Regression

Symbolic regression produces an equation, i.e., a correlation, by applying different mathematical operations to the training data. This methodology produces a good fitting of a mathematical expression, but the drawback is that the algorithm can generate correlations without any physical meaning. Therefore, the use of Chapman-Enskog to normalize the data in the form D^*/D_0^* allows the obtained correlation to be semi-empirical and the correlations obtained from SR simply quantify the deviation of D^* with respect to the value of this property for a reference dilute gas of hard spheres.

The correlations obtained from the SR of the vapour, liquid, and supercritical states, as well as for the entire data set are presented in Table II. The performance of these correlations is shown in the parity plots presented in Figure 3, and the summary of the values of AARD and R^2 are presented in Table I. The equation for the self-diffusion coefficient of the vapour phase, Equation 10, is the simplest of all models and indicates that D^* is approximately 65% of the value of D_0^*, which makes sense since D_0^* is derived for fluids of infinite dilution. Moreover, the correlation does not include any dependence on the cohesive parameter α, meaning that the expression is independent of the intermolecular potential. Despite the simplicity, this expression is sufficient to predict the self-diffusion coefficient to great accuracy as observed in Figure 3(a), in which both training and testing data sets lie on top of the diagonal in the parity plot. The AARD for this model is 3.5% and $R^2 = 0.995$. The simplicity of the equation is due to the small range in D^*/D_0^* that is observed in the vapour phase.

Contrary to the equation obtained for the vapour phase, the one for the liquid phase, Equation 11, is complex and introduces new additional terms. One important distinction, however, is the appearance of the cohesive parameter α in the expression suggesting the relevance of the shape of the intermolecular potential in this high-density region and at low temperatures where particle-particle correlations are important. This expression performs equally well when compared to

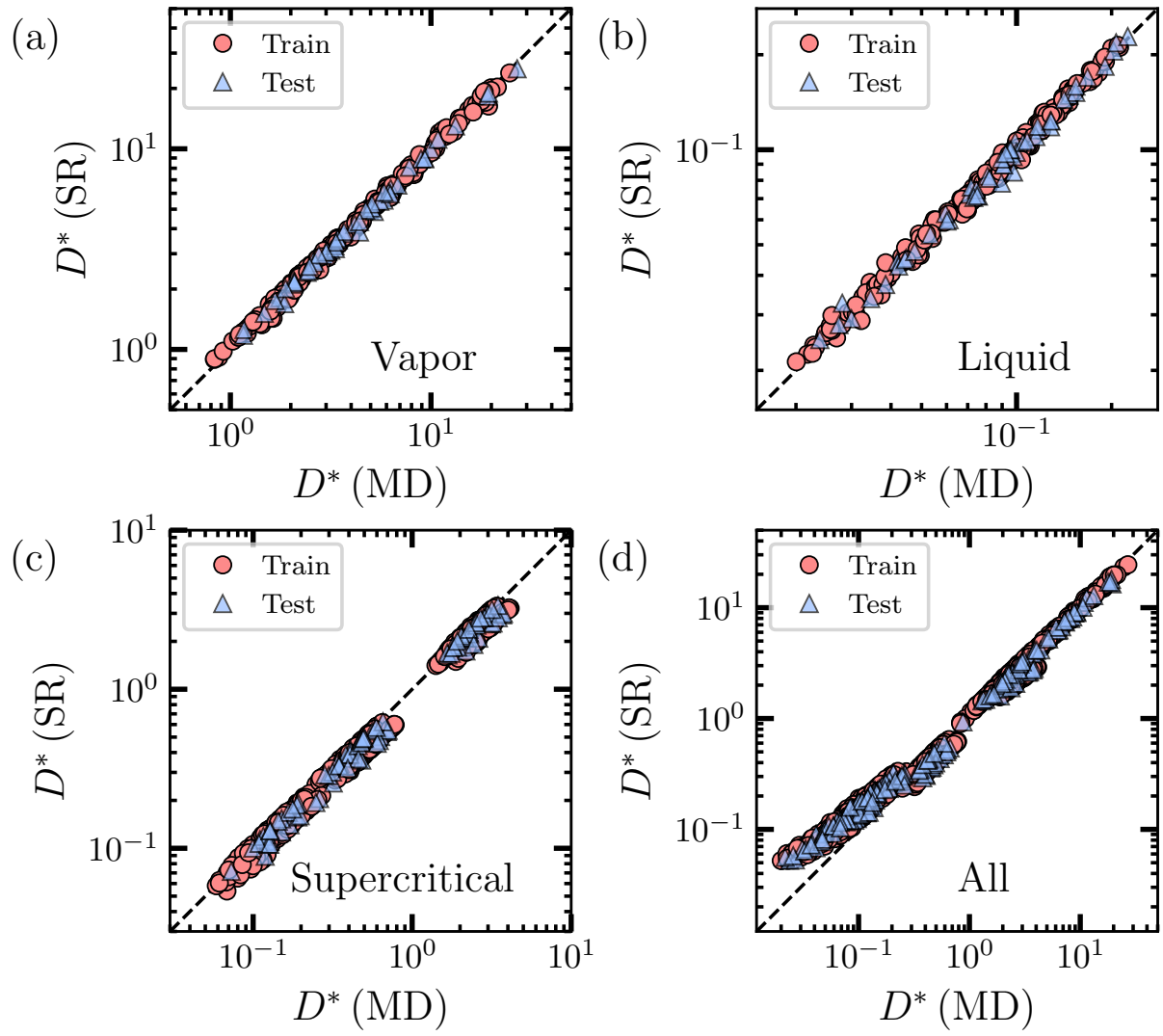

FIG. 3. Parity plots describing the performance of the equations obtained from symbolic regression (SR) for the description of the self-diffusion coefficient for three different regions in the phase diagram, as well as for the entire data set as a whole. The results correspond to (a) subcritical vapour, (b) subcritical liquid, (c) supercritical fluid, and (d) all states. Training data are shown in red circles while the testing data are shown in blue triangles. D^* (MD) denotes the self-diffusion coefficients obtained from MD simulations, while D^* (SR) denotes the predictions using symbolic regression.

the KNN algorithm but underperforms in comparison to the ANN algorithm when considering the performance metrics. The strength of the SR algorithm is its interpretability, as the obtained equation is very simple to implement, whereas the inner working of the ANN and KNN algorithms are 'black boxes'.

TABLE II. Semi-empirical equations obtained from the symbolic regression of the self-diffusion coefficient D^* and the viscosity η^* at different states.

Property	State	Equation	
D^*	Vapour phase	$\frac{D^*}{D_0^*} = 0.649$	(10)
D^*	Liquid phase	$\frac{D^*}{D_0^*} = [T^*(1+\exp(-\alpha)-0.694]\,\frac{\exp(-\rho^*)}{\sqrt{\alpha}}$	(11)
D^*	Supercritical phase	$\frac{D^*}{D_0^*} = \sqrt{(0.664-\rho^{*2})\sqrt{\frac{2}{3}T^*}}$	(12)
D^*	All states	$\frac{D^*}{D_0^*} = \exp(-\rho^*/T^*)\exp(-\lvert 0.273\log(\log(1/\rho^*))\rvert)$	(13)
η^*	Liquid phase	$\frac{\eta^*}{\eta_0^*} = \frac{\exp(\rho^*)\,[\exp(\rho^*)+\log(\alpha)]}{T^*-0.405}$	(14)
η^*	Liquid phase	$\eta^* = 0.892\alpha\rho^*\left(1+\frac{\rho^*}{T^*(T^*-0.381)}\right)$	(15)

The performance of the equation obtained for the supercritical phase is shown in Figure 3(c) and the expression is presented in Equation 12. This expression is relatively simple compared to the equation for the liquid phase. Due to the lack of very high-density state points in the data set, this equation is only valid for $\rho^* < \sqrt{0.664}$ to ensure that the square root function produces a real value. The lack of high-density state points is also a contributing factor to the absence of any dependence on the cohesive parameter α. In comparison to the algorithms discussed previously for this data set, SR is outperformed by the KNN algorithm, but exhibiting nearly equivalent performance to ANN.

Finally, the performance of the SR equation obtained for the entire data set can be observed in Figure 3(d) and is given by Equation 13. This expression exhibits a very good value of R^2, but a poor AARD. While it is clear that the equation is unphysical, it has very good predictive power, having a value of R^2 equivalent to the black box ML methods. Much of the error in this equation comes from the overprediction of low values of D^* as can be seen in Figure 3(d), which is the location of the self-diffusion coefficients obtained from the liquid phase simulations. As the SR equation for all phases does not take into account the cohesive parameter α of the potential, the liquid self-diffusion coefficients are not predicted accurately, where the value of the cohesion parameter has a larger influence on the diffusion coefficient due to the large proximity of particles and low temperature of the system. Additionally, the liquid phase data only makes 25% of the full data set, which makes the effects of the cohesion parameter more difficult to be captured for

the SR algorithm. The lack of accuracy at low self-diffusion coefficients is balanced by the high precision of high-self diffusion coefficients, which are not as susceptible to changes in the cohesive parameter of the potential.

It is also important to note that depending on the value of the initial seed set in the random number generator used in the SR algorithm, there is a larger variability of the expression obtained (except for the vapour phase). The expressions presented in Table II are only a subset of the expressions found by the algorithm.

B. Shear viscosity

As for the case of the self-diffusion coefficient, the ANN and KNN algorithms, as well as SR, have been applied to represent the shear viscosity η^* of particles interacting *via* the Mie potential. However, only the liquid region is analysed in this work as it is the main region of interest for fluid flow applications. The performance of both ANN and KNN algorithms is shown in Figure 4(a). Similar to the case of the self-diffusion coefficients, the ANN exhibits a better performance to predict the shear viscosity of the liquid region with AARD of 3.3% and $R^2 = 0.988$. In contrast, the AARD and R^2 for the KNN algorithm are 2.8% and 0.977, respectively. This difference in predicting power between the measurements shows that KNN is more accurate for low values of η^*, while ANN is more accurate for high values. Interestingly, both methods improve when they are trained using non-normalised data, that is, when the models are train with respect to D^* instead of trained with respect to D^*/D_0^*. This enhacement in their performance is shown in Figure 4(b), especially for ANN, with AARD$= 2.21\%$ and $R^2 = 0.996$. KNN receives a smaller improvement with AARD$= 2.6\%$ and $R^2 = 0.977$.

For the case of the SR applied to the shear viscosity, the performance of the methodology also follows a similar behaviour as in the case of the self-diffusion coefficient. The semi-empirical equation obtained for SR has been obtained by fitting η^*/η_0^*, Equation 14, and the results are presented in Figure 5(a) and in Table I. It is clear from these results that the SR performs worse than the KNN and ANN algorithms, in terms of both R^2 and AARD. Furthermore, the equation is complex and has a dependency on the cohesion parameter α, mirroring the equation obtained for the predictions of the self-diffusion coefficient for the liquid phase. This shows that at high densities there is a more pronounced effect of the exact inter-particle potential and, hence, the requirement for the inclusion of cohesion parameter becomes more apparent. For higher values of

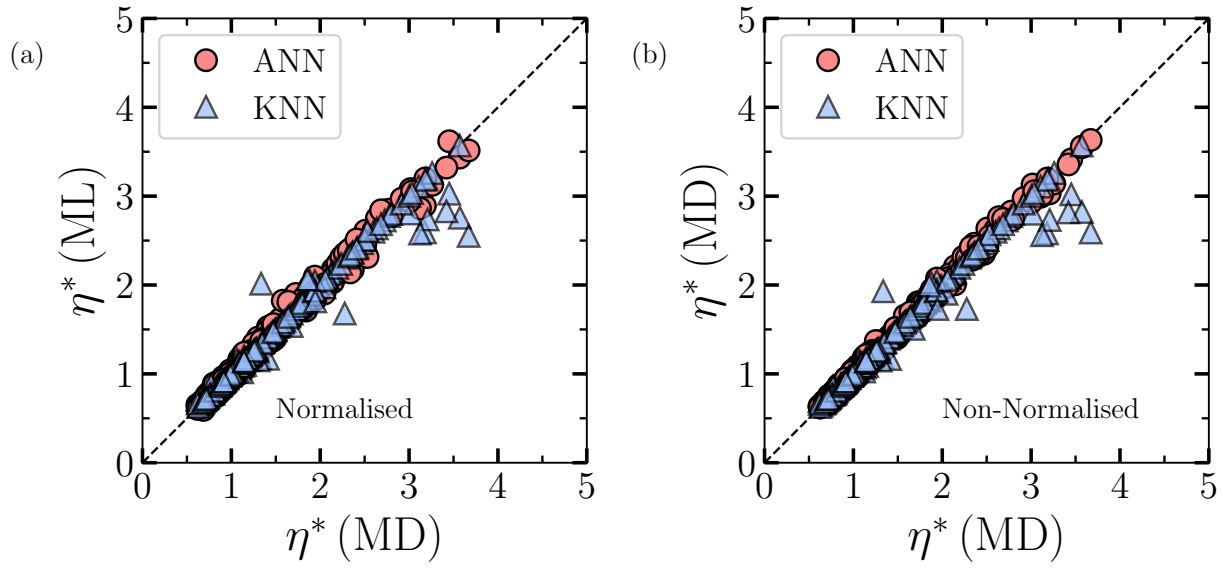

FIG. 4. Parity plot describing the performance of the ANN (red circles) and KNN (blue triangles) on the testing set of the shear viscosity η^* of the liquid phase for (a) normalised and (b) non-normalised (right). η^* (MD) denotes the shear viscosity obtained from MD simulations, while η^* (ML) denotes the predictions using machine learning.

η, it is observed that there is an increase in deviations of the data from the parity line. While this semi-empirical equation performs well, it can be observed in Figure 5(a) that the slope of the data at low values of η^* also deviate from the main diagonal, implying that there are missing factors in the equation. For comparison, the SR algorithm has also been trained with respect to η^* instead of η^*/η_0^* to obtain a fully empirical equation, given by Equation 15, for the shear viscosity and the results are presented in Figure 5(b). This equation exhibits an AARD=3.8% and a $R^2 = 0.987$, which shows that the empirical model has slightly better performance in both metrics compared to the semi-empirical equation, with the value of R^2 outperforming the value for KNN and equaling the values of ANN. The shape of the points on the fully empirical equation also follows the parity line more closely, especially at low viscosity values.

C. Application to real fluids

In this section, the application of the ML models to predict the self-diffusion coefficients of real fluids is demonstrated. The selected systems correspond to krypton Kr, methane CH_4 and carbon dioxide CO_2. These systems have been chosen as they can be represented approximately

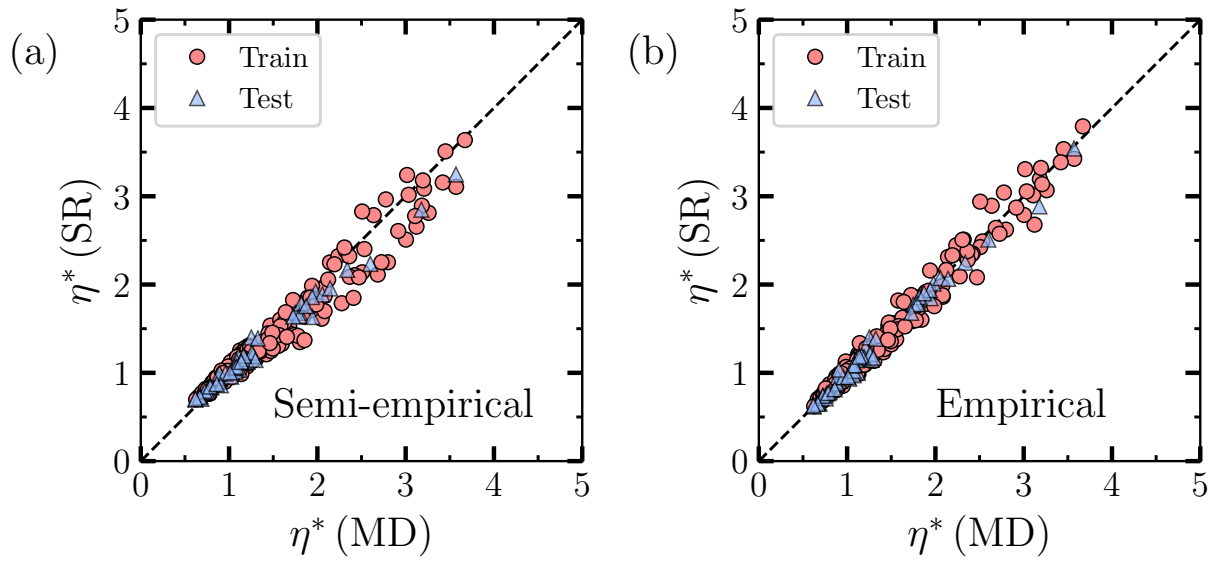

FIG. 5. Parity plots describing the performance of the equations obtained from symbolic regression for the description of the shear viscosity η^* for the liquid phase. The results correspond to (a) the semi-empirical equation (Equation 14) and (b) the empirical equation (Equation 15). Training data are shown in red circles while testing data are shown in blue triangles. η^* (MD) denotes the shear viscosity obtained from MD simulations, while η^* (SR) denotes the predictions using symbolic regression.

by a single coarse-grained sphere and exhibit a wide variety of intermolecular interactions that the Mie potential can capture. The representation of CO_2 as a single coarse-grained sphere has been previously demonstrated[71] and is used in this work. To find the best Mie potential parameters that describe the other fluids (Kr, CH_4), the highly accurate SAFT-VR Mie EoS is used[9]. This equation is based on a high-order Barker and Henderson perturbation theory and is able to represent with high accuracy the vapour-liquid equilibria, including an excellent representation of the critical region.

The methodology to predict the self-diffusion coefficients of these systems is as follows. First, experimental vapour-liquid equilibria and vapour pressures have been obtained from the NIST Chemistry WebBook[88] and fitted to the SAFT-VR Mie EoS to find the intermolecular parameters of the Mie potential, i.e. σ, ε, n and m, that best represent the vapour and liquid saturation densities, as well as the vapour pressure. Using the calculated cohesive parameter α from the optimized exponents, the ML algorithms optimized for all the fluid regions in the phase diagram can be deployed to predict the self-diffusion coefficient at any temperature and density. Finally,

TABLE III. Results for the optimized Mie intermolecular parameters ε, σ, n and m obtained using the SAFT-VR Mie EoS[9]. The cohesive parameter α is obtained from Equation 2.

System	$\varepsilon/(k_B/\mathrm{K})$	$\sigma/\mathrm{Å}$	n	m	α
Kr	176.34	3.663	14.28	5.98	0.795
CH_4	160.53	3.754	14.15	5.98	0.800
CO_2	353.55	3.741	23.0	6.66	0.358

the optimized values of σ and ε are used to convert from reduced units to real units and the results are compared with experimental data collected recently by Allers *et al.*[63].

The optimized Mie intermolecular parameters obtained from the fitting of the SAFT-RV EoS, and the values used for CO_2 from Avendaño *et al.*[71], are presented in Table III, and the corresponding prediction of the saturated vapour and liquid densities are shown in Figures 6(a,c,e) for Kr, CH_4, and CO_2, respectively. As can be observed in these Figures, the SAFT-VR Mie provides an excellent representation of the saturated densities when compared to the experimental results reported in NIST[88]. Using the parameters reported in Table III, the self-diffusion coefficients for the three substances are predicted using, for example, the optimized ANN model and can be compared with available experimental data[63]. The results of this comparison are presented in the parity plots shown in Figures 6(b,d,f) for Kr, CH_4, and CO_2, respectively. The calculated metrics AARD and R^2 for the three substances are also reported in Table IV. The predictions using KNN and SR are also presented in the SI.

For KNN and ANN the predictions are equivalent for all three fluids. The predictions are best for CO_2 in both metrics. When studying the parity plots presented in Figure 6, the parity plot of CO_2 looks the most accurate over the range of values studied, while for CH_4 and Kr there are strong deviations at low values of D^*. These deviations may be attributed directly to the lack of training data in very dense liquids, where the intermolecular forces play a larger role. Additionally, the coarse-graining applied to the fluid particles treats them as perfect spheres. For example., CO_2 is neither spherical nor can be represented by simple dispersion forces due to the strong quadrupolar moments of the molecule. The same effect has been reported by Aimoli *et al.* in their study of transport properties of CO_2 using the same coarse-grained model[71,73]. The deviations at low values of D^* are more pronounced in the SR predictions, where the cohesion parameter is ignored entirely.

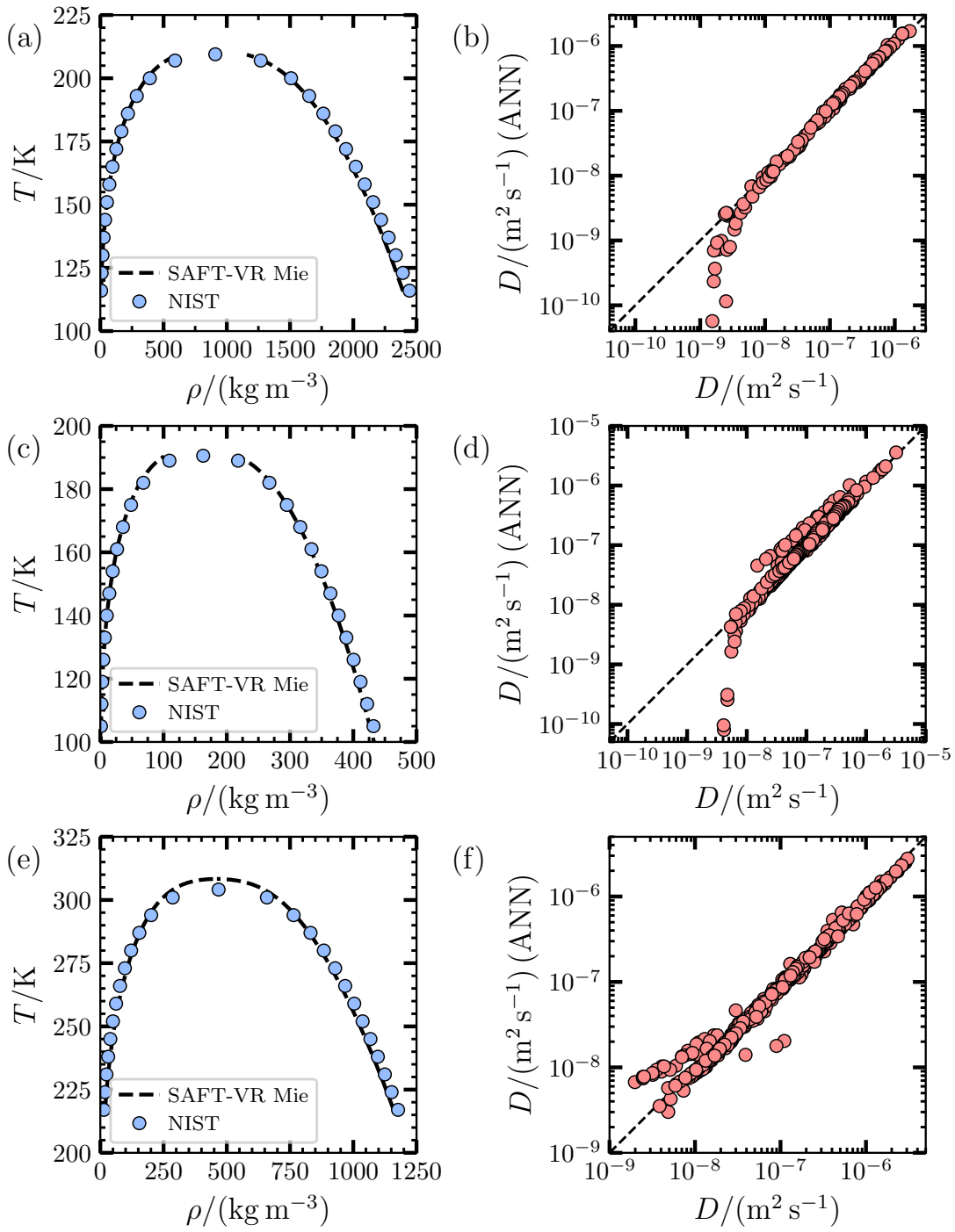

FIG. 6. Results for the vapour-liquid equilibria and self-diffusion coefficients of (a,b) krypton Kr, (c,d) methane CH_4, and (e,f) carbon dioxide CO_2. In figures (a, c, d), the blue circles correspond to the experimental results obtained from NIST[88] while the dashed curves are the predictions using the SAFT-VR Mie Eos[9] using the parameters reported in Table III. The parity plots shown in Figures (b,d,f) show the comparison of the prediction of the ANN developed in this work with the experimental results collected by Allers *et al.*[63].

TABLE IV. The results for the metrics AARD and R^2 obtained from the prediction of the self-diffusion coefficient using the ML methods optimized for all states.

	KNN		ANN		SR	
System	AARD	R^2	AARD	R^2	AARD	R^2
Kr	26.6%	0.970	23.4%	0.968	45.8%	0.906
CH_4	24.7%	0.970	24.9%	0.971	40.6%	0.902
CO_2	22.9%	0.972	22.7%	0.977	24.1%	0.944

IV. CONCLUSION

In summary, the ability of three different ML algorithms to predict transport properties of spheres interacting *via* the Mie potential has been investigated. The main transport property studied in this work is the self-diffusion coefficient, which has been obtained from the calculations of the mean-squared displacement from MD simulations. It has been found that KNN and ANN perform equivalently over all the fluid phases considered, with KNN having slightly better performance in general, especially in terms of AARD. This is attributed to the interpolation used in the KNN method, which allows for a more accurate prediction of small D^* values. The region studied in the supercritical regime was trained on a biased data set, resulting in reduced performance from all methods employed. The SR model performs slightly worse, in general, than the methods discussed previously. The worst performance from SR is the high AARD in the all-phase data set, where the lack of the cohesion parameter in the SR expression leads to reduced performance in the low D^* region. Additionally, for all the ML methods used, viscosity was better predicted when training without the use of η_0.

The equations obtained from applying the SR algorithm have shown an unexpected result, where the cohesive parameter, which defines the potential's fluid phase diagram, is absent for all the fluid phase regions except for the liquid phase. The prediction of the shear viscosity has reaffirmed the comparisons for the performance of the algorithms, as the same trends have been observed. To increase the generalization and applicability of the ML models, the all-phase models have been applied to predict the self-diffusion coefficients of krypton, methane and carbon dioxide. To that end, the Mie potential parameters for each fluid have been calculated using SAFT-VR Mie equation of state using available vapour-liquid experimental data. These parameters have been

used along with the previously presented models to predict the self-diffusion coefficient of the three selected systems. While a good overall agreement has been observed, the results have shown some deviations that are attributed to the lack of training data at very high liquid density regime as well as the simplicity of single-sphere coarse-grained representation.

V. SUPPLEMENTARY INFORMATION

The supplementary information contains a discussion on the uncertainties in the calculation of the transport properties, heat maps representing the accuracy of the models in the phase space, prediction of self-diffusion coefficients of Lennard-Jones particles from published literature, and the predicted results of the self-diffusion coefficient for Kr, CH_4, and CO_2 using the three ML models.

VI. ACKNOWLEDGEMENTS

This work was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) via an Industrial Cooperative Award in Science & Technology (ICASE) co-funded by IBM, project ID 2327699 - EP/T517689/1. The authors would like to acknowledge the assistance given by Research IT and the use of the Computational Shared Facility at The University of Manchester. This work was also supported by the Hartree National Centre for Digital Innovation, a collaboration between STFC and IBM. A.P. is supported by a "Maria Zambrano Senior" distinguished researcher fellowship, financed by the European Union within the NextGenerationEU program and the Spanish Ministry of Universities.

REFERENCES

[1]E. Hendriks, G. M. Kontogeorgis, R. Dohrn, J.-C. de Hemptinne, I. G. Economou, L. F. Zvilnik, and V. Vesovic, Ind. Eng. Chem. Res. **49**, 11131 (2010).

[2]K. E. Gubbins, Y.-C. Liu, J. D. Moore, and J. C. Palmer, Phys. Chem. Chem. Phys. **13**, 58 (2011).

[3]B. E. Poling, J. M. Prausnitz, and J. O'Connell, *The properties of gases and liquids*, 5th ed. (McGraw-Hill Professional, New York, NY, 2001).

[4]J. O. Valderrama, Ind. Eng. Chem. Res. **42**, 1603 (2003).

[5]M. Michelsen and J. Mollerup, *Thermodynamic Models: Fundamentals & Computational Aspects* (Tie-Line Publications, 2004).

[6]W. G. Chapman, K. E. Gubbins, G. Jackson, and M. Radosz, Ind. Eng. Chem. Res. **29**, 1709 (1990).

[7]A. Gil-Villegas, A. Galindo, P. J. Whitehead, S. J. Mills, G. Jackson, and A. N. Burgess, J. Chem. Phys. **106**, 4168 (1997).

[8]J. M. Prausnitz, R. N. Lichtenthaler, and E. G. de Azevedo, *Molecular thermodynamics of fluid-phase equilibria* (Pearson Education, Philadelphia, PA, 1998).

[9]T. Lafitte, A. Apostolakou, C. Avendaño, A. Galindo, C. S. Adjiman, E. A. Müller, and G. Jackson, J. Chem. Phys. **139**, 154504 (2013).

[10]D. A. McQuarrie, *Statistical Mechanics*, Harper's chemistry series (Longman, London, England, 1976).

[11]S. E. Quiñones-Cisneros, C. K. Zéberg-Mikkelsen, and E. H. Stenby, Fluid Phase Equilib. **169**, 249 (2000).

[12]D. R. Reichman and P. Charbonneau, J. Stat. Mech. **2005**, P05013 (2005).

[13]A. Mulero, ed., *Theory and simulation of hard-sphere fluids and related systems*, 2008th ed., Lecture notes in physics (Springer, Berlin, Germany, 2008).

[14]D. Frenkel and B. Smit, *Understanding Molecular Simulation*, 2nd ed. (Elsevier, 2002).

[15]M. P. Allen and D. J. Tildesley, *Computer simulation of liquids* (Oxford University Press, London, England, 2017).

[16]P. R. Sassiat, P. Mourier, M. H. Caude, and R. H. Rosset, Anal. Chem. **59**, 1164 (1987).

[17]D. Chandler, J. Chem. Phys. **62**, 1358 (1975).

[18]R. J. Speedy, Mol. Phys. **62**, 509 (1987).

[19]R. Speedy, F. Prielmeier, T. Vardag, E. Lang, and H.-D. Lüdemann, Mol. Phys. **66**, 577 (1989).

[20]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).

[21]F. Demmel, D. Szubrin, W.-C. Pilgrim, and C. Morkel, Phys. Rev. B **84**, 014307 (2011).

[22]H. Liu and C. M. Silva, Ind. Eng. Chem. Res. **36**, 246 (1997).

[23]N. M. Blagoveshchenskii, A. G. Novikov, and V. V. Savostin, Phys. Solid State **56**, 120 (2014).

[24]N. Blagoveshchenskii, A. Novikov, A. Puchkov, V. Savostin, and O. Sobolev, EPJ Web of Conferences **83**, 02018 (2015).

[25]G. V. Kharlamov and S. V. Zhilkin, J. Phys.: Conf. Ser. **899**, 052009 (2017).

[26]Y. A. Aljeshi, M. B. M. Taib, and J. P. M. Trusler, Int. J. Thermophys. **42**, 140 (2021).

[27]C. Corral-Casas, L. Gibelli, M. K. Borg, J. Li, S. F. K. Al-Afnan, and Y. Zhang, Phys. Fluids **33**, 082009 (2021).

[28]R. L. Rowley and M. M. Painter, Int J Thermophys **18**, 1109 (1997).

[29]R. Laghaei, A. E. Nasrabad, and B. C. Eu, J. Phys. Chem. B **109**, 5873 (2005).

[30]L. V. Woodcock, AIChE Journal **52**, 438 (2006).

[31]X. Wang, S. Ramírez-Hinestrosa, J. Dobnikar, and D. Frenkel, Phys. Chem. Chem. Phys. **22**, 10624 (2020).

[32]S. E. Quiñones-Cisneros, C. K. Zéberg-Mikkelsen, J. Fernández, and J. García, AIChE J. **52**, 1600 (2006).

[33]A. S. de Wijn, V. Vesovic, G. Jackson, and J. P. M. Trusler, J. Chem. Phys. **128**, 204901 (2008).

[34]F. Llovell, R. M. Marcos, and L. F. Vega, J. Phys. Chem. B **117**, 8159 (2013).

[35]S. Becker, E. Devijver, R. Molinier, and N. Jakse, Phys. Rev. E **105**, 045304 (2022).

[36]W. F. Reinhart, A. W. Long, M. P. Howard, A. L. Ferguson, and A. Z. Panagiotopoulos, Soft Matter **13**, 4733 (2017).

[37]S. Dasetty, I. Coropceanu, J. Portner, J. Li, J. J. de Pablo, D. Talapin, and A. L. Ferguson, Mol. Syst. Des. Eng. **7**, 350 (2022).

[38]J. O'Leary, R. Mao, E. J. Pretti, J. A. Paulson, J. Mittal, and A. Mesbah, Soft Matter **17**, 989 (2021).

[39]J. L. McDonagh, W. C. Swope, R. L. Anderson, M. A. Johnston, and D. J. Bray, Polym. Int. **70**, 248 (2021).

[40]E. Boattini, M. Dijkstra, and L. Filion, J. Chem. Phys **151**, 154901 (2019).

[41]M. Spellings and S. C. Glotzer, AIChE J. **64**, 2198 (2018).

[42]A. Fabrizio, B. Meyer, and C. Corminboeuf, J. Chem. Phys. **152**, 154103 (2020).

[43]C. Dietz, T. Kretz, and M. H. Thoma, Phys. Rev. E **96**, 011301 (2017).

[44]J. L. McDonagh, N. Nath, L. De Ferrari, T. van Mourik, and J. B. O. Mitchell, J. Chem. Inf. Model. **54**, 844 (2014).

[45]J. L. McDonagh, D. S. Palmer, T. v. Mourik, and J. B. O. Mitchell, J. Chem. Inf. Model. **56**, 2162 (2016).

[46]J. L. McDonagh, T. van Mourik, and J. B. O. Mitchell, Mol. Inform. **34**, 715 (2015).

[47]L. Joss and E. A. Müller, J. Chem. Educ. **96**, 697 (2019).

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

[49]A. S. Alshehri, A. K. Tula, F. You, and R. Gani, AIChE J. **68**, e17469 (2022).

[50]K. K. Yalamanchi, V. C. O. van Oudenhoven, F. Tutino, M. Monge-Palacios, A. Alshehri, X. Gao, and S. M. Sarathy, J. Phys Chem. A **123**, 8305 (2019).

[51]G. B. Goh, C. Siegel, A. Vishnu, N. O. Hodas, and N. Baker, IEEE , 1340 (2018).

[52]V. Mann, K. Brito, R. Gani, and V. Venkatasubramanian, Fluid Ph. Equilibria **561**, 113531 (2022).

[53]A. Tihic, G. M. Kontogeorgis, N. von Solms, M. L. Michelsen, and L. Constantinou, Ind. Eng. Chem. Res. **47**, 5092 (2007).

[54]K. Hansen, F. Biegler, R. Ramakrishnan, W. Pronobis, O. A. von Lilienfeld, K.-R. Müller, and A. Tkatchenko, J. Phys. Chem. Lett. **6**, 2326 (2015).

[55]J. Schmidt, M. R. G. Marques, S. Botti, and M. A. L. Marques, Npj Comput. Mater. **5**, 83 (2019).

[56]A. Abbasi Moud, Colloids Interface Sci. Commun. **47**, 100595 (2022).

[57]M. Ihme, W. T. Chung, and A. A. Mishra, Prog. Energy Combust. Sci **91**, 101010 (2022).

[58]Y. Elbaz, D. Furman, and M. Caspary Toroker, Adv. Funct. Mater. **30**, 1900778 (2020).

[59]A. Abbasi and R. Eslamloueyan, Chemometr. Intell. Lab. Syst. **132**, 39 (2013).

[60]X. Zhao, T. Luo, and H. Jin, Ind. Eng. Chem. Res. **61**, 8542 (2022).

[61]N. Melzi, L. Khaouane, S. Hanini, M. Laidi, Y. Ammi, and H. Zentou, J. Appl. Mech. Tech. Phys. **61**, 207 (2020).

[62]J. Zhou, S. Chupradit, K. Ershov, W. Suksatan, H. Abdulameer Marhoon, M. Alashwal, S. Ghazali, M. Algarni, and A. El-Shafay, J. Mol. Liq. **353**, 118808 (2022).

[63]J. P. Allers, F. H. Garzon, and T. M. Alam, Phys. Chem. Chem. Phys. **23**, 4615 (2021).

[64]J. P. S. Aniceto, B. Zêzere, and C. M. Silva, J. Mol. Liq. **326**, 115281 (2021).

[65]J. P. Allers, J. A. Harvey, F. H. Garzon, and T. M. Alam, J. Chem. Phys. **153**, 034102 (2020).

[66]K. Papastamatiou, F. Sofos, and T. E. Karakasidis, AIP Advances **12**, 025004 (2022).

[67]C. J. Leverant, J. A. Harvey, and T. M. Alam, J. Phys. Chem. Lett. **11**, 10375 (2020).

[68]T. M. Alam, J. P. Allers, C. J. Leverant, and J. A. Harvey, J. Chem. Phys. **157**, 014503 (2022).

[69]G. Mie, Ann. Phys. (Berl.) **11**, 657 (1903).

[70]X. He, W. Shinoda, R. DeVane, and M. L. Klein, Mol. Phys. **108**, 2007 (2010).

[71]C. Avendaño, T. Lafitte, A. Galindo, C. S. Adjiman, G. Jackson, and E. A. Müller, J. Phys. Chem. B **115**, 11154 (2011).

[72]C. Avendaño, T. Lafitte, C. S. Adjiman, A. Galindo, E. A. Müller, and G. Jackson, J. Phys. Chem. B **117**, 2717 (2013).

[73]C. G. Aimoli, E. J. Maginn, and C. R. A. Abreu, J. Chem. Phys. **141**, 134101 (2014).

[74]J. C. Shelley, M. Y. Shelley, R. C. Reeder, S. Bandyopadhyay, and M. L. Klein, J. Phys. Chem. B **105**, 4464 (2001).

[75]J. J. Potoff and D. A. Bernard-Brunel, J. Phys. Chem. B **113**, 14725 (2009).

[76]N. S. Ramrattan, C. Avendaño, E. A. Müller, and A. Galindo, Mol. Phys. **113**, 932 (2015).

[77]K. S. Pitzer, J. Chem. Phys. , 583 (1939).

[78]E. Helfand and S. A. Rice, J. Chem. Phys. , 1642 (1960).

[79]J. A. Barker and D. Henderson, Rev. Mod. Phys. **48**, 587 (1976).

[80]W. G. Hoover, Phys. Rev. A **31**, 1695 (1985).

[81]A. P. Thompson, H. M. Aktulga, R. Berger, D. S. Bolintineanu, W. M. Brown, P. S. Crozier, P. J. in 't Veld, A. Kohlmeyer, S. G. Moore, T. D. Nguyen, R. Shan, M. J. Stevens, J. Tranchida, C. Trott, and S. J. Plimpton, Comp. Phys. Comm. **271**, 108171 (2022).

[82]R. Kubo, J. Phys. Soc. Jpn. **12**, 570 (1957).

[83]M. S. Green, J. Chem. Phys. **22**, 398 (1954).

[84]Y. Zhu, X. Lu, J. Zhou, Y. Wang, and J. Shi, Fluid Phase Equilib. **194-197**, 1141 (2002).

[85]F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, J. Mach. Learn. Res. **12**, 2825 (2011).

[86]T. Stephens, "Gplearn: Genetic programming in python, with a scikit-learn inspired API," (2015).

[87]J. Šlepavičius, C. Avendaño, B. O. Conchúir, and A. Patti, Phys. Rev. E **106**, 014604 (2022).

[88]P. Linstrom, "NIST Chemistry WebBook, NIST Standard Reference Database 69," (1997).