Nicolas Hayer
Matrix Completion Methods for the Prediction of Thermodynamic Properties of Mixtures
Scientific Report Series Volume 53
2025

Scientific Report Series
Laboratory of Engineering Thermodynamics (LTD)
RPTU Kaiserslautern
P.O. Box 3049
67663 Kaiserslautern
Germany

ISSN 2195-7606
ISBN 978-3-944433-52-3

Matrix Completion Methods for the Prediction of Thermodynamic Properties of Mixtures

Vom Fachbereich Maschinenbau und Verfahrenstechnik
der Rheinland-Pfälzischen Technischen Universität
Kaiserslautern-Landau
zur Verleihung des akademischen Grades

Doktor-Ingenieur (Dr.-Ing.)

genehmigte

Dissertation

von

M.Sc. Nicolas Hayer
aus Wittlich

Dekan:	Prof. Dr. rer. nat. Roland Ulber
Berichterstatter:	Prof. Dr.-Ing. Hans Hasse
	Prof. Dr.-Ing. Fabian Jirasek
	Prof. Dr. rer. nat. Stephan Mandt

Tag der mündlichen Prüfung: 21.03.2025

D 386

Danksagung

Die vorliegende Arbeit, die ich während meiner Zeit am Lehrstuhl für Thermodynamik (LTD) der RPTU Kaiserslautern angefertigt habe, wäre ohne die Unterstützung vieler Menschen nicht möglich gewesen. Dafür möchte ich mich an dieser Stelle herzlich bedanken.

Bei Prof. Hans Hasse bedanke ich mich für die hervorragende fachliche Betreuung und die große wissenschaftliche Freiheit auf dem Weg zur Promotion. Ebenso gilt mein besonderer Dank Prof. Fabian Jirasek für die zahlreichen fachlichen Diskussionen, das stets freundliche Miteinander und die spürbare Wertschätzung.

Einen wesentlichen Beitrag zu meiner Promotion haben die engagierten Studierenden geleistet, die ich während meiner Zeit am LTD betreuen durfte – herzlichen Dank dafür. Des Weiteren gebührt ein großer Dank meinen zahlreichen Kolleginnen und Kollegen, die einen unverzichtbaren Beitrag zur Entstehung dieser Arbeit geleistet haben. Ganz gleich, ob Sekretariat, Laborteam oder Leidensgenossinnen auf dem Weg zur Promotion, für den Zusammenhalt und eure Unterstützung bin ich sehr dankbar. Gemeinsam haben wir die vielen Traditionen des LTD gepflegt und neue ins Leben gerufen. Neben fachlichen Diskussionen und gemeinsamen Projekten hatte auch der Spaß stets seinen Platz und ich bin froh, viele von euch heute zu meinen guten Freundinnen zählen zu dürfen.

Ein ganz besonderer Dank gilt all jenen, die mein Leben abseits der Promotion unvergleichlich schön gemacht haben: meinen Freunden aus Schulzeit und Studium für den gemeinsamen Spaß und die regelmäßigen Urlaube, meinen Eltern für ihre bedingungslose Unterstützung und meinem Bruder Fabi für einfach alles.

Kaiserslautern, Mai 2025

Nicolas Hayer

Abstract

The accurate prediction of thermodynamic properties is pivotal for chemical engineering as experimental data are scarce. While established physics-based methods face limitations in prediction accuracy and scope, emerging machine learning approaches, such as matrix completion methods (MCMs), offer promising alternatives. MCMs exploit the fact that experimental data for binary mixtures can be represented as elements of a sparse matrix, with rows and columns corresponding to the components that make up the mixture. Hence, MCMs can be used for closing the gaps in these matrices. In the present thesis, new methods for predicting thermodynamic properties of mixtures are developed that combine probabilistic MCMs with established physical methods. The resulting hybrid methods yield significantly improved predictions for key properties of binary mixtures, such as Henry's law constants, activity coefficients at infinite dilution, and diffusion coefficients at infinite dilution, even when only limited experimental training data are available. In addition to predicting mixture properties directly, this thesis demonstrates that MCMs can also be applied to the pair-interaction parameters of physical group-contribution (GC) methods, which suffer from incomplete and improvable parameter sets limiting their applicability and accuracy. By using MCMs to infer the pair-interaction parameters, a comprehensive and consistent parameter set can be generated. This approach extends the applicability of widely used GC methods such as UNIFAC and modified UNIFAC (Dortmund), ultimately increasing their scope, predictive power, and robustness.

Kurzfassung

Eine präzise Vorhersage thermodynamischer Eigenschaften ist in der chemischen Industrie von zentraler Bedeutung, da experimentelle Daten nur in begrenztem Umfang verfügbar sind. Während etablierte physikalische Methoden hinsichtlich Vorhersagegenauigkeit und Anwendungsbreite an ihre Grenzen stoßen, bieten neue Ansätze des maschinellen Lernens, insbesondere sogenannte Matrixvervollständigungsmethoden (Matrix Completion Methods, MCMs), vielversprechende Alternativen. MCMs schließen vorhandene Datenlücken, indem sie ausnutzen, dass experimentelle Daten für binäre Mischungen als Elemente einer spärlich besetzten Matrix dargestellt werden können, deren Zeilen und Spalten den Komponenten der Mischungen entsprechen. In der vorliegenden Dissertation werden neue Methoden zur Vorhersage thermodynamischer Eigenschaften von Mischungen entwickelt, die probabilistische MCMs mit etablierten physikalischen Methoden kombinieren. Diese hybriden Ansätze erzielen selbst bei geringer Verfügbarkeit experimenteller Trainingsdaten hohe Vorhersagegenauigkeiten für wichtige Eigenschaften binärer Mischungen wie Henry-Konstanten, Aktivitätskoeffizienten bei unendlicher Verdünnung und Diffusionskoeffizienten bei unendlicher Verdünnung. Neben der direkten Vorhersage von Mischungseigenschaften zeigt diese Dissertation, dass MCMs auch zur Parametrisierung von Gruppenbeitragsmethoden (GC-Methoden) verwendet werden können, die durch unvollständige Parametersätze eingeschränkt sind. Durch den Einsatz von MCMs zur Ermittlung der Paarwechselwirkungsparameter kann ein vollständiger und konsistenter Parametersatz erzeugt werden. Dieser Ansatz erweitert den Anwendungsbereich weit verbreiteter GC-Methoden wie UNIFAC und modified UNIFAC (Dortmund) und erhöht zusätzlich ihre Vorhersagegenauigkeit und Robustheit.

Contents

List of Symbols

Latin symbols

A_{ij}, B_{ij}, C_{ij}	system-specific parameters
a_{mn}, b_{mn}, c_{mn}	pair-interaction parameters
b_i^u	solute bias
b_j^v	solvent bias
D	Fickian diffusion coefficient
$Đ$	Maxwell-Stefan diffusion coefficient
$\boldsymbol{D}$	data matrix
G^{E}	excess Gibbs energy
g^{E}	molar excess Gibbs energy
H_{ij}	Henry's law constant of the solute i in the solvent j
h^{abs}	enthalpy of absorption
h^{E}	excess enthalpy
I	number of solutes
J	number of solvents
K	latent dimension
M	molar mass
m, n	numbers of rows and columns of the matrix
N_i	number of components i
N_{MG}	number of main groups
N_{min}	minimal number of data points
N_{obs}	number of observed entries
P	parachor
P_{ij}	system-specific parameter
p	probability density function
p	pressure
p_i^{s}	vapor pressure of pure component i
$\bar{p}_m(\sigma)$	modified σ-profile
Q_k	surface area of subgroup k
R_k	volume of subgroup k

r_{obs}	observation ratio
S	similarity score
T	temperature
T_{c}	critical temperature
U_{mn}	group-interaction energy between main group m and n
$\boldsymbol{U}$	solute feature matrix
$\boldsymbol{u}_i$	feature verctor of solute i
$\boldsymbol{V}$	solvent feature matrix
v	molar volume
$\boldsymbol{v}_j$	feature verctor of solvent j
w	weighting factor
x_i	mole fraction of component i in the liquid phase
y_i	mole fraction of component i in the vapor phase
$\boldsymbol{Z}$	data matrix

Subscripts and superscripts

0	reference
$*$	preliminary
∞	at infinite dilution
A	surface area
abs	absorption
CB	component bias
c	at the critical point
E	excess
exp	experimental
i	component
k	subgroup
L	likelihood
MG	main group
m, n	main group
min	minimal
obs	observation
P	prior
p	polar regions in the σ-profile
pred	predicted
σ	surface charge distribution

Greek symbols

$\boldsymbol{\beta}$, $\boldsymbol{\theta}$	feature verctors of structural groups
Γ	thermodynamic factor
γ	activity coefficient
Δ	difference
ε	experimental noise
η	dynamic viscosity
λ	scale parameter of the Cauchy distribution
μ	mean
ξ	threshold
ϱ	density
σ	standard deviation
σ	screening charge density
Ψ	parameter of the UNIFAC model

Abbreviations

ADVI	Automatic Differentiation Variational Inference
ANN	artificial neural network
COSMO-SAC model	COSMO segment activity coefficient model
COSMO-RS	conductor-like screening model for real solvents
DDB	Dortmund Data Bank
DIPPR	Design Institute for Physical Properties
DOE	design of experiment
ELBO	Evidence lower bound
EoS	equations of state
GC	group-contribution
GNN	graph neural network
IQR	interquartile range
LLE	liquid-liquid equilibrium
LV	latent variable
MAE	mean absolute error
MAPE	mean absolute percentage error
MCM	matrix completion method
MCMC	Markov chain Monte Carlo
ML	machine learning
Mod. UNIFAC (DO)	modified UNIFAC (Dortmund)

MSE	mean squared error
PSRK EoS	predictive Soave-Redlich-Kwong equations of state
QSPR	quantitative structure-property relationships
rMSE	root mean squared error
SBM	similarity-based method
SEGWE model	Stokes-Einstein Gierer-Wirtz estimation model
SLE	solid-liquid equilibrium
SMARTS	SMILES arbitrary target specification
SMILES	simplified molecular-input line-entry system
UNIFAC	universal quasichemical functional group activity coefficients
VI	variational inference
VLE	vapor-liquid equilibrium

1 Introduction

Knowledge of the thermodynamic properties of mixtures is crucial for designing and optimizing industrial processes. However, experimental data are often lacking due to the high cost and complexity of measurements. As a result, predictive methods are essential to estimate these properties.

While physical methods for predicting mixture properties are well established, they often only have a limited scope and lack accuracy. To overcome these limitations, new machine learning (ML) approaches have recently been explored [1–3]. In the present thesis, matrix completion methods (MCMs) from ML, which are well established in recommender systems [4–6], are used to improve the prediction of thermodynamic properties.

The basic idea of an MCM is simple: sparse data for binary mixtures can be conveniently stored in a matrix. Predicting the missing entries in such a matrix constitutes a matrix completion problem. A variety of MCMs exist to address this [7], but a particularly effective and straightforward approach is matrix factorization. Thereby, the data matrix is decomposed into two lower-dimensional matrices containing so-called features. During training, these features are learned, and their product reconstructs the completed matrix, as detailed in Chapter 2.

For the prediction of thermodynamic properties of mixtures, MCMs can be applied in two key ways, both of which are used in this thesis. First, they can directly complete sparse experimental data matrices. Second, they can complete matrices of binary interaction parameters of physical models. This embedding of MCMs in a physical framework yields powerful hybrid models with increased applicability, e.g., by enabling extrapolation to unstudied conditions and components.

First applications of MCMs to predict thermodynamic properties have focused on activity coefficients γ_{ij}^{∞} of solutes i infinitely diluted in solvents j, which describe liquid-phase non-ideality [8–11]. This thesis builds on this approach and introduces improved methods for predicting γ_{ij}^{∞}. Additionally, it applies MCMs to predict Henry's law constants H_{ij}, which describe the gas solubility, and diffusion coefficients at infinite dilution D_{ij}^{∞}, which characterize molecular motion in mixtures. In this thesis, different training strategies of MCMs have been tested, ranging from training only on sparse experimental data to training on extended databases augmented with synthetic data from various sources.

These synthetic data are derived from physical methods and a new prediction method developed in this thesis that is based on component similarity derived from quantum-chemical descriptors.

Physical group-contribution (GC) methods are widely employed for the prediction of thermodynamic properties but suffer from incomplete interaction parameter matrices, limiting their accuracy and applicability. In this thesis, these limitations are overcome by embedding MCMs in GC methods to complete the essential pair-interaction parameter matrices. This novel approach is applied to the most successful GC methods, UNIFAC [12] and modified UNIFAC (Dortmund) [13], which model the excess Gibbs energy of mixtures and can predict related properties. Extensive training sets of experimental data for activity coefficients and, in the case of modified UNIFAC (Dortmund), excess enthalpies, taken from the Dortmund Data Bank, were used. The proposed methods can be easily implemented in existing software packages and can be easily updated as new experimental data become available or adapted to specific industrial needs.

The present thesis is organized as follows: In Chapter 2, the concept of matrix completion by matrix factorization is explained and a brief introduction into Bayesian modeling is given, forming the basis for all MCMs presented in Chapters 4 and 5. Chapter 3 presents the similarity-based method (SBM), a novel approach for predicting thermodynamic properties of binary mixtures that is based on quantifying the pairwise similarity of components by comparing their quantum-chemical descriptors. As an example, the SBM is applied to the prediction of missing entries in a matrix containing isothermal activity coefficients at infinite dilution (γ_{ij}^{∞}). In Chapter 4, Bayesian MCMs for the prediction of essential thermodynamic properties in binary mixtures (H_{ij}, γ_{ij}^{∞}, and D_{ij}^{∞}) are presented. In Chapter 5, the application of Bayesian MCMs to the prediction of missing pair-interaction parameters in the most successful GC models in chemical engineering, UNIFAC and modified UNIFAC (Dortmund), is discussed.

2 Bayesian Matrix Factorization

Matrix completion methods (MCMs) aim to estimate missing entries in a sparse data matrix $\boldsymbol{Z}$. A widely used approach is matrix factorization, which models each matrix entry Z_{ij}, where i denotes the row and j denotes the column, as the dot product of two vectors, $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$ [14]:

$$Z_{ij} = \boldsymbol{u}_i \cdot \boldsymbol{v}_j + \varepsilon_{ij} \tag{1}$$

Here, $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$ are the model parameters, specific for row i and column j, respectively, which are often referred to as features. These features are fitted to the observed data in matrix $\boldsymbol{Z}$ to minimize the residuals ε_{ij} during the training of the model. This approach relies on correlated entries in the matrix. The MCM learns these correlations and captures them through the features.

In a Bayesian approach, each data point (Z_{ij}) and each feature ($\boldsymbol{u}_i$, $\boldsymbol{v}_j$) are modeled as random variables with associated probability distributions. Two key probability distributions thereby define a Bayesian model: the prior and the likelihood.

The prior constitutes a probability distribution over the model's features before any data are observed, representing the initial assumptions about the features. Thus, the prior contains a priori information about the features prior to the actual training step.

The likelihood determines how the features relate to the observed data by specifying a probability distribution over the data conditioned on the features. This distribution also influences how the observations update the features during training. The choice of the probability distribution used as the likelihood can be considered a model hyperparameter, influencing the model's flexibility, generalization, and sensitivity to noise and outliers. For example, while the normal distribution is often chosen for its simplicity and stability, the Cauchy distribution, with its heavy tails, is preferred in noisy data sets to provide robustness against outliers and extreme values.

Bayesian inference aims to determine the so-called posterior of the features, which is a probability distribution over the features and integrates both prior assumptions (through the prior) and evidence from observed data (through the likelihood). Formally, Bayes' theorem states:

$$p(\boldsymbol{u}_i, \boldsymbol{v}_j \,|\, Z_{ij}) \propto p(Z_{ij} \,|\, \boldsymbol{u}_i, \boldsymbol{v}_j) \cdot p(\boldsymbol{u}_i, \boldsymbol{v}_j) \tag{2}$$

where $p(Z_{ij} \,|\, \boldsymbol{u}_i, \boldsymbol{v}_j)$ is the likelihood and $p(\boldsymbol{u}_i, \boldsymbol{v}_j)$ is the prior. Thus, the posterior $p(\boldsymbol{u}_i, \boldsymbol{v}_j \,|\, Z_{ij})$ represents the updated beliefs about the features after incorporating both the initial assumptions and the empirical evidence.

However, the exact computation of the posterior is often intractable due to the lack of a closed-form solution or excessive computational cost [15]. To address these issues, two prominent strategies are commonly used to achieve Bayesian inference: Markov chain Monte Carlo (MCMC) sampling [16, 17] and variational inference (VI) [15, 18]. MCMC methods generate samples from the posterior distribution by constructing a Markov chain that sequentially explores the parameter space. Although MCMC provides a flexible approach, its iterative nature and requirement for many samples can lead to significant computational costs, especially when dealing with very large data sets. In contrast, VI transforms the inference problem into an optimization task: it posits a family of simpler, tractable probability distributions and finds the member of this family that is closest to the true posterior with respect to a chosen divergence measure. This often results in faster inference and improved scalability compared to MCMC, but at the cost of introducing an approximation bias.

Throughout this thesis, VI is applied to approximate the posterior for the Bayesian MCMs. Specifically, in Chapters 4 and 5.1, Gaussian mean-field VI is used with the Automatic Differentiation Variational Inference (ADVI) algorithm [19], implemented in the Stan probabilistic programming language [20]. MatlabStan, which allows seamless integration of Stan code into MATLAB scripts [21–23], is thereby used. However, training hybrid group-contribution methods on hundreds of thousands of experimental data points (cf. Chapter 5.2) requires extensive computing power and, thus, GPU parallelization capabilities, which are not supported by Stan. In these cases, Pyro [24], a Python-based probabilistic programming language supported by PyTorch, is used to perform stochastic VI under the mean-field assumption [15]. Each latent feature is approximated by a normal variational distribution, and during VI, the evidence lower bound (ELBO) is maximized using the Adam optimizer [25] with a learning rate of 0.15, ensuring efficient and scalable training.

All MCMs presented in this thesis follow the Bayesian approach described in this chapter, with one exception: the similarity-based method (SBM) introduced in Chapter 3. The SBM is unique not only in its inference strategy. In general, MCMs can also be categorized by the type of information they use. Content-based filtering methods incorporate direct item-specific details (e.g., properties of rows and columns) [26], whereas

collaborative-filtering methods rely solely on observed matrix entries and uncover latent patterns to predict missing values [4, 27]. According to this classification, the SBM is content-based, while all other MCMs in this thesis fall into the collaborative-filtering category.

3 Similarity-Based Imputation

3.1 Introduction

Thermodynamic properties of mixtures are fundamental for the design and optimization of processes. In this chapter, a novel approach is described for predicting properties of binary mixtures based on *similarities* between components. This novel similarity-based method (SBM) is built on the fundamental assumption that similar components exhibit similar properties (*similia similibus solvuntur*), making component similarities highly informative inputs for predictive thermodynamic models.

Molecular similarity is commonly used in computational chemistry and pharmaceutical research for database searching and component selection in high-throughput screening. The goal of these applications is to find components that exhibit a behavior that is similar to that of a reference component with desired properties. This is achieved by identifying similar substructures or calculating overall similarity measures, resulting in a list of the most similar molecules in the database and, ultimately, guiding drug discovery and optimization. To perform these pairwise molecular comparisons, a molecular representation of the components and a method to evaluate the similarity based on these representations are required. Various approaches have been proposed for this purpose in the literature, each with its own merits and limitations [28, 29].

The most common molecular representations for similarity searches are molecular fingerprints, which encode structural information into bit vectors, such as the presence of specific functional groups [29, 30]. Analyzing fingerprint similarities is computationally efficient, as it only involves comparing bit strings. The Tanimoto coefficient is the most popular metric for assessing fingerprint similarity [30–32]. Other molecular representations for assessing similarity include molecular graphs, molecular descriptor vectors, SMILES, SMARTS, and pharmacophores [28, 29, 33]. Molecular descriptors based on quantum-chemical charge distribution calculations, such as σ-profiles [34], are rarely used to assess similarities in pharmaceutical research, despite their potential [35, 36].

While the idea of using similarities is implicitly at the heart of many models for predicting thermodynamic properties for unstudied systems, the proposed similarity-based method (SBM) exploits that idea based on a measure of similarity directly.

Among the thermodynamic properties of mixtures, the activity coefficient is particularly significant since it quantifies the non-ideality of liquid mixtures, which is essential for accurately modeling reaction and phase equilibria [37]. A highly informative limiting case is the activity coefficient γ_{ij}^{∞} of a solute i infinitely diluted in a solvent j, as many mixture properties can be predicted based on the knowledge of the limiting activity coefficients. However, despite their importance, experimental data for γ_{ij}^{∞} are scarce, even in comprehensive databases for thermophysical properties such as the Dortmund Data Bank [38], due to the high cost and time required for their measurement [39, 40]. Consequently, reliable prediction methods are essential.

Activity coefficients are usually calculated from models of the excess Gibbs energy G^{E}. Predictions for binary mixtures, for which no data are available, can be obtained from group-contribution methods, namely UNIFAC [12, 41] and modified UNIFAC (Dortmund) [13, 42], or using the COSMO-RS approach [34, 43, 44], which is based on quantum-chemical component descriptors, the σ-profiles. Open-source versions of COSMO-RS include COSMO-SAC [45, 46] and COSMO-SAC-dsp [47]. The σ-profiles describe the screening charge density of a molecule embedded in an electrically conductive continuum by a probabilistic distribution $p(\sigma)$ across the molecule's surface segments, where σ is the charge of the segment [34].

In addition to these physical prediction methods, new machine learning (ML) methods and hybrid models that combine physics with ML have been developed recently [1, 2]. These methods include graph neural networks (GNN) [48], transformer models [49], and matrix completion methods (MCM) [8–10]. Additionally, many ML methods have been developed to predict activity coefficients over the entire concentration range, which could also be applied to the special case of activity coefficients at infinite dilution [50–54].

The SBM is applied here to predict activity coefficients at infinite dilution γ_{ij}^{∞} in binary mixtures. The SBM thereby relies on two sources of information: a novel similarity measure S_{mn} between two components m and n and available experimental data for γ_{ij}^{∞}. The similarity measure S_{mn} is based on a comparison of σ-profiles of the pair of components and used to screen the experimental database, identifying γ_{ij}^{∞} values from similar mixtures that are then used for predictions by imputation. The developed SBM is benchmarked with modified UNIFAC (Dortmund) [13], COSMO-SAC [46], and COSMO-SAC-dsp [47] as three well established physics-based methods for predicting γ_{ij}^{∞}. It is emphasized that the SBM for predicting γ_{ij}^{∞} is an example; the approach is generic and can be transferred to any other binary property.

3.2 Database

Experimental data on activity coefficients at infinite dilution in binary mixtures, γ_{ij}^{∞}, were obtained from the Dortmund Data Bank (DDB) [38]. In the preprocessing step, all data sets containing undefined components or labeled as "poor quality" by the DDB were discarded. The focus was restricted to binary mixtures at a temperature of $T = 298.15 \pm 1$ K. If multiple measurements existed for the same binary mixture, the median of these values was adopted. For scaling purposes, the logarithmic activity coefficients, $\ln \gamma_{ij}^{\infty}$, were used throughout this chapter.

The proposed SBM uses σ-profiles obtained from quantum-chemical COSMO calculations to calculate the similarity between two components. In this chapter, the σ-profiles were taken from the open-source database provided by Bell et al. [55], which features results for 2,261 different components. Components not available in this database were excluded from the data set.

Finally, for evaluating the model using leave-one-out analysis, at least two experimental data points were required for each solute and solvent; therefore, data for which this condition was violated were removed. The final data set is visualized in Fig. 1 and comprises 3,568 data points for γ_{ij}^{∞}, covering 221 solutes and 198 solvents.

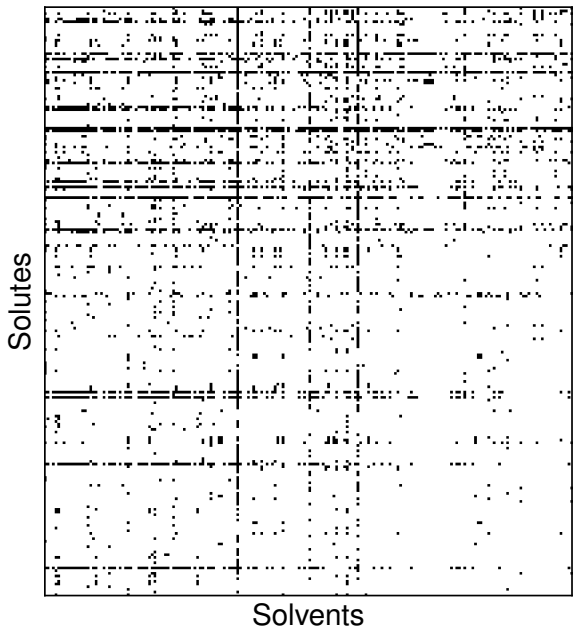


Figure 1: Matrix representing the experimental data on logarithmic activity coefficients at infinite dilution $\ln \gamma_{ij}^{\infty}$ for binary mixtures at 298.15±1 K from the DDB [38] after preprocessing (see text). Experimental data are available for 3,568 binary mixtures, constituting about 8% of all possible combinations of the considered 221 solutes and 198 solvents.

3.3 Similarity-Based Method

3.3.1 Similarity Score

Here, a novel similarity score S_{mn} between two components m and n based on quantum-chemical COSMO calculations is introduced. The score S_{mn} is scaled such that its values range from 0 (highly dissimilar components) to 1 (highly similar components) and consists of two contributions, as also indicated in Fig. 2: the similarity based on surface charge distributions S^{σ}_{mn} and the similarity of the surface area S^{A}_{mn} as it is also used in the COSMO method; S^{σ}_{mn} and S^{A}_{mn}, which are described in detail in the following, are also defined to range from 0 to 1. The final similarity score S_{mn} is obtained from a weighted sum of S^{σ}_{mn} and S^{A}_{mn}:

$$S_{mn} = w_{\sigma} \cdot S^{\sigma}_{mn} + (1 - w_{\sigma}) \cdot S^{A}_{mn} \tag{3}$$

where w_{σ} is the weighting factor that controls the relative importance of the surface charge distribution similarity compared to the surface area similarity.

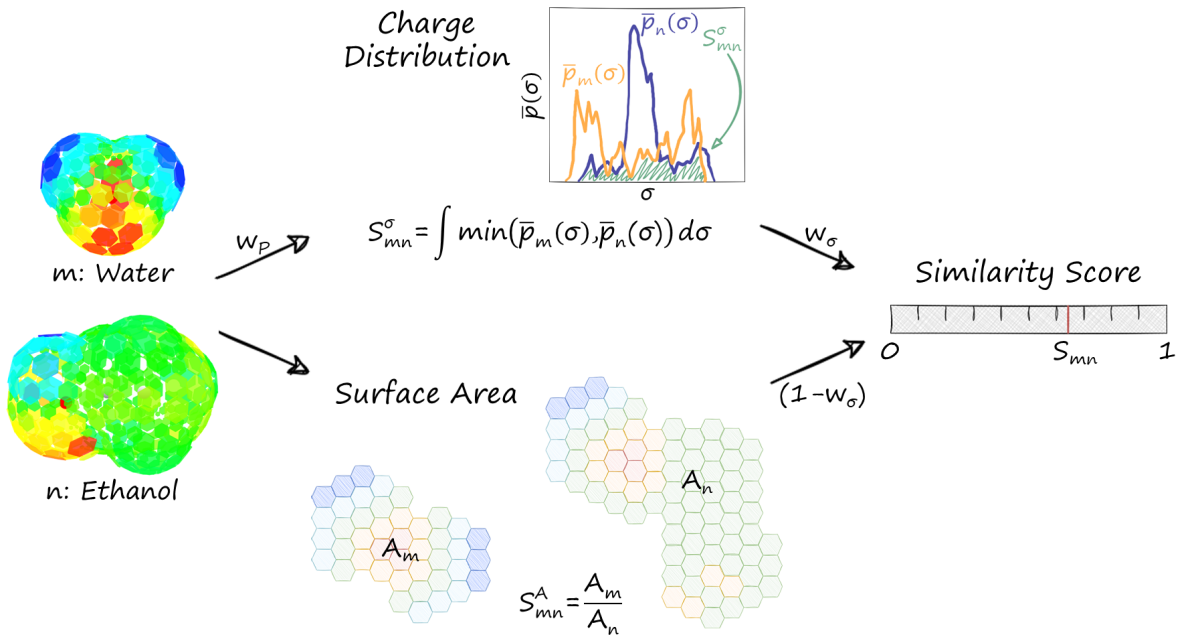


Figure 2: Schematic depiction of calculating the similarity between two components (water and ethanol in this example) as proposed in this chapter. The final similarity score S_{mn} is composed of two contributions: a similarity based on charge distribution S^{σ}_{mn} and a size similarity derived from the surface areas S^{A}_{mn}, which are combined in a weighted sum.

The size similarity S^A_{mn} is defined as the cavity surface area A of the smaller molecule divided by the one of the larger molecule:

$$S^A_{mn} = \begin{cases} \frac{A_m}{A_n}, & \text{if } A_m < A_n \\ \frac{A_n}{A_m}, & \text{if } A_m > A_n \end{cases} \tag{4}$$

For the similarity of the surface charge distributions S^σ_{mn}, the overlapping proportion of the σ-profiles of the two components is used, which is calculated using discrete bins for σ via:

$$S^\sigma_{mn} = \sum_{k=1}^{N_\sigma} \min\left(\bar{p}_m(\sigma_k), \bar{p}_n(\sigma_k)\right) \tag{5}$$

where $\bar{p}_m(\sigma_k)$ and $\bar{p}_n(\sigma_k)$ are modified σ-profiles, preprocessed as described in the following. All σ-profiles are given here in a discretized version with σ being divided into $N_\sigma = 51$ bins ranging from -0.025 $\mathrm{e\,Å^{-2}}$ to 0.025 $\mathrm{e\,Å^{-2}}$ with a constant step size of 0.001 $\mathrm{e\,Å^{-2}}$. These values will be referred to as σ_k for $k = 1, \ldots, 51$. Thus, $p_m(\sigma_k)$ is the fraction of the surface area of the component m associated with the screening charge density σ_k.

The σ-profiles are modified by introducing w_P, which is applied to control the weight on the polar regions in the σ-profiles by being either 0 (no influence) or 2 (more focus on polar regions):

$$p^*_m(\sigma_k) = p_m(\sigma_k) \cdot (10^3 \sigma_k)^{w_\mathrm{P}} \tag{6}$$

By setting $w_\mathrm{P} = 2$, the similarity calculation emphasizes charge-dense regions, which can be crucial in cases where the behavior of the components is mainly determined by polar interactions.

In the case of $w_\mathrm{P} = 2$, the resulting $p^*_m(\sigma_k)$ does not integrate to 1. Therefore, it is normalized again:

$$p^{**}_m(\sigma_k) = \frac{p^*_m(\sigma_k)}{\sum_{k=1}^{N_\sigma} p^*_m(\sigma_k)} \tag{7}$$

In the final processing step, a potential issue associated with discretized σ-profiles is addressed. Specifically, when calculating the similarity score by comparing the σ-profiles of two molecules bin-wise, small shifts in σ can prevent the detection of structurally similar molecules. Therefore, a moving average with a sliding window of width 2 (corresponding to 0.002 $\mathrm{e\,Å^{-2}}$) is applied to all profiles to increase the robustness:

$$\bar{p}_m(\sigma_k) = \frac{p^{**}_m(\sigma_{k-1}) + p^{**}_m(\sigma_k)}{2} \tag{8}$$

The resulting σ-profiles $\bar{p}_m(\sigma_k)$ are used for calculating the similarity of the surface charge distributions S^{σ}_{mn} (see Eq. (5)). Together with the similarity of the surface area S^{A}_{mn} (see Eq. (4)), the final similarity score S_{mn} is calculated (see Eq. (3)).

The two introduced weights w_σ (in Eq. (3)), and w_P (in Eq. (6)) are hyperparameters, which were determined by a grid search. The value ranges of the hyperparameters explored in the grid search are detailed in the "Studied Model Variants" section. In addition to these two weights, other modifications to the calculation of S^{σ}_{mn} (e.g., emphasizing hydrogen-bonding surface segments) and of S^{A}_{mn} (e.g., including component volume) were tested in preliminary studies, but showed no significant impact on the performance of the SBM and were, therefore, discarded.

3.3.2 Prediction of Activity Coefficients

In this section, it is explained how the similarity score defined in the previous section is applied for predicting activity coefficients at infinite dilution $\ln \gamma^{\infty}_{ij}$ in unstudied mixtures, where, basically, the $\ln \gamma^{\infty}_{ij}$ is just an example for a property of a binary mixture. The respective method introduced is called the similarity-based method (SBM). The central idea of the SBM is to find mixtures similar to the unstudied mixture that is of interest but for which experimental data on $\ln \gamma^{\infty}_{ij}$ are available. The activity coefficient in the unstudied mixture, $\ln \gamma^{\infty,\mathrm{pred}}_{ij}$, is then predicted simply by arithmetically averaging the corresponding experimental values $\ln \gamma^{\infty,\mathrm{exp}}_{ij}$ of all similar mixtures.

Here, a *similar mixture* is defined as one with the same solute i (or the same solvent j) but a different solvent n (a different solute m) for which the similarity score S_{nj} (S_{mi}) is higher than a predefined threshold ξ, i.e., $S_{nj} > \xi$ ($S_{mi} > \xi$). Consequently, at least one similar mixture for which an experimental data point is available must be in the training set to make a prediction. As a result, there will always be a trade-off when applying the SBM: increasing the threshold value ξ will increase the accuracy, but it will lower the range of applicability. Vice versa, decreasing the value of ξ will increase the range of applicability but decrease the accuracy.

A leave-one-out approach [56] was applied to assess the SBM to guarantee true predictions. These predictions are also used in comparing the SBM results with the physical benchmark models, which results in a bias in favor of the physical models, as they were very likely also trained with at least some of the data considered here. All calculations of the present chapter were carried out using Matlab [22].

3.3.3 Studied Model Variants

The SBM described in the previous sections uses two weights, w_σ and w_P, in calculating the similarity score S_{mn}. These weights were varied in a grid search to explore their effects on model performance. Specifically, w_σ was varied from 0 to 1 in increments of 0.1, while w_P was set to either 0 or 2. This setup resulted in 22 distinct SBM configurations, each representing a different approach to the S_{mn} calculation. The goal of this grid search was to identify the SBM (i.e., weight combination) that performs best for two, often conflicting, objectives: optimizing the accuracy in predicting $\ln \gamma_{ij}^{\infty}$ in terms of mean absolute error (MAE) and maximizing the scope, i.e., the number of predictable mixtures.

The best-performing SBM, according to these objectives, retains one further adjustable hyperparameter: the threshold ξ, which allows users to balance the trade-off between accuracy and scope. Increasing ξ typically results in more accurate predictions but limits the number of predictable data points since higher similarities are demanded for making predictions. Conversely, lowering ξ increases the number of predictable points but reduces the predictive accuracy since data for less similar components are used for the predictions. To assess the impact of ξ, it was varied from 0.5 to 1 in increments of 0.01 for each of the 22 SBM configurations.

3.4 Results and Discussion

3.4.1 Overall Performance of Different Similarity-Based Methods

Fig. 3 shows the predictive accuracy in terms of the MAE of the predicted $\ln \gamma_{ij}^{\infty}$ over the number of predictable data points N from the data set for all tested SBM variants (by varying the weights and ξ).

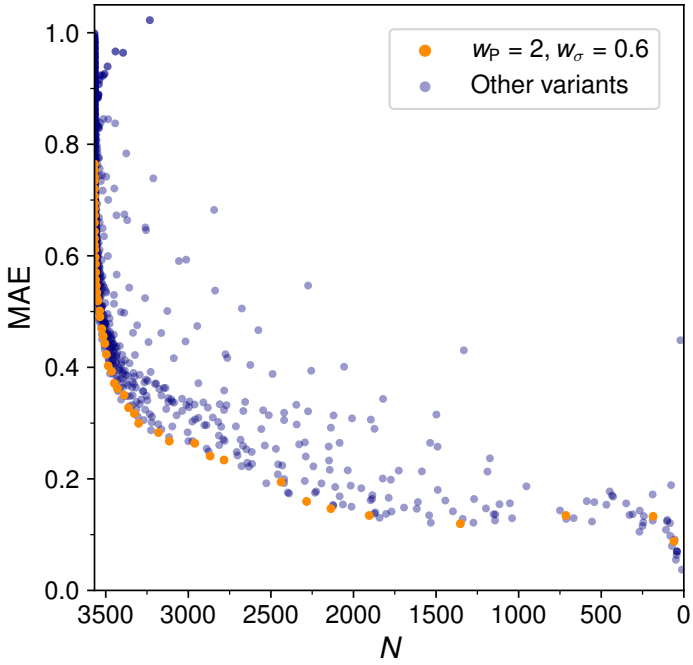


Figure 3: Mean absolute error (MAE) of the predicted $\ln \gamma_{ij}^{\infty}$ from the leave-one-out analysis over the number of predictable experimental data points N for all tested SBM variants. The results of the best-performing SBM (as specified with the weights w) are highlighted in orange.

The model variants in Fig. 3 scatter across a broad range of MAE and N, underscoring the substantial impact of the selected hyperparameters on model performance. This range highlights the inherent trade-off between predictive accuracy and scope, representing a Pareto optimization problem. In such cases, a solution is considered Pareto-optimal if no feasible solution improves at least one objective without worsening another. Here, certain hyperparameter combinations yield Pareto-optimal SBM variants that achieve maximum accuracy for a given scope and vice versa. The set representing all Pareto-optimal solutions is called the Pareto front.

One particular SBM (with variable ξ) consistently lies on or near the Pareto front, highlighted in orange in Fig. 3. This "best" SBM, defined by w_σ = 0.6 and w_P = 2,

requires only the final tuning of ξ by users to achieve a near-optimal solution tailored to their specific preferences.

The balanced value of $w_\sigma = 0.6$ in the best SBM indicates that components must share similarities in both surface charge distribution and surface area to exhibit comparable values of $\ln \gamma_{ij}^{\infty}$. Furthermore, $w_\mathrm{P} = 2$ emphasizes the importance of a similar surface charge distribution in the polar regions of the components for similar $\ln \gamma_{ij}^{\infty}$.

Figs. A.1 and A.2 in Appendix A show further analysis of specific hyperparameter choices. The similarity scores calculated by the best SBM can be used to identify the most similar components for a target component, as exemplified in Appendix A (cf. Tables A.2 and A.3).

3.4.2 Comparison to Physical Benchmark Models

The best-performing SBM ($w_\sigma = 0.6$, and $w_\mathrm{P} = 2$) selected in the grid search is further evaluated in the following by comparison against the state-of-the-art physical benchmark methods for predicting activity coefficients: modified UNIFAC (Dortmund) [13], COSMO-SAC [46], and COSMO-SAC-dsp [47]. As shown in Fig. 4, the methods are compared using the MAE and the scope regarding the number of predictable data points N in the data set. Additionally, the deviations of the predictions from the experimental data are plotted in histograms for the SBM with $\xi = 0.93$, modified UNIFAC (Dortmund), and COSMO-SAC-dsp. Modified UNIFAC (Dortmund) has some extreme outliers, which were excluded from the MAE calculations in Fig. 4. A detailed analysis of these outliers can be found in Appendix A, cf. Table A.1.

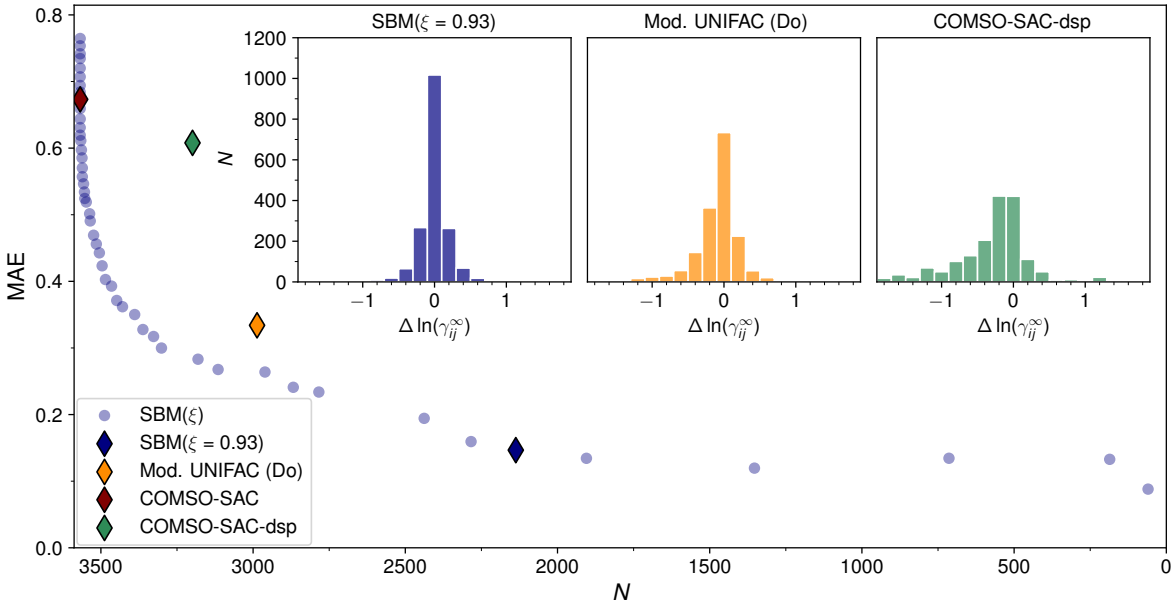


Figure 4: Mean absolute error (MAE) of the best-performing SBM (with varied thresholds ξ), modified UNIFAC (Dortmund), COSMO-SAC, and COSMO-SAC-dsp for the prediction of $\ln \gamma_{ij}^{\infty}$ over the number of predictable experimental data points N. Insets provide histograms of the deviations of the predictions with the SBM with $\xi = 0.93$, modified UNIFAC (Dortmund), and COSMO-SAC-dsp from the experimental data, considering only mixtures that all three methods can describe. The shown interval in the histograms contains 99.9% (SBM), 96.7% (modified UNIFAC (Dortmund)), and 96.9% (COSMO-SAC-dsp) of the relevant 1,748 data points.

In Fig. 4, the scope is discussed regarding the number of predictable data points from the experimental database. An additional discussion of scope, in terms of the filling level of the entire matrix as shown in Fig. 1, i.e., the predictions for mixtures for which no experimental data are available, is provided in Appendix A.

First of all, it is evident from Fig. 4 that for the physical models, there is also a trade-off between the scope of the method and its accuracy. COSMO-SAC-dsp is more accurate than COSMO-SAC, but in its current parameterization [55], it is not applicable to components containing certain halogens due to missing parameters for the dispersion part, resulting in a slightly smaller scope. Both COSMO variants have a larger scope than modified UNIFAC (Dortmund) but achieve less accurate results.

Compared to each physical benchmark method, one can always find an SBM variant (by varying ξ) that outperforms it in terms of prediction accuracy and scope by selecting an appropriate threshold. Specifically, at $\xi = 0.62$, the SBM can, like COSMO-SAC, predict all binary systems in the database but achieves a better MAE (0.62 compared to 0.67). At $\xi = 0.85$, the SBM has a broader scope than COSMO-SAC-dsp ($N = 3,301$ compared to $N = 3,199$) and achieves a better MAE (0.30 compared to 0.61). Similarly, the SBM

with ξ = 0.87 has a broader scope than modified UNIFAC (Dortmund) (N = 3,115 compared to N = 2,987) and achieves a better MAE (0.27 compared to 0.33).

For the following analysis, the threshold is fixed to ξ = 0.93. While this value is, in principle, arbitrary, the resulting model can predict more than half of the available experimental data in the database with relatively high predictive accuracy. The deviations of the predictions from the experimental data for each method are also represented as histograms in Fig. 4. Most of the predictions of the SBM with ξ = 0.93 show deviations from experimental data smaller than ±0.1, which is within the typical range of experimental uncertainty of $\ln \gamma_{ij}^{\infty}$, underscoring the high quality of the predictions that can be obtained with the proposed model.

To further analyze the performance of the best-performing SBM, the respective objectives (MAE and N) are plotted over the threshold ξ, as shown in Fig. 5.

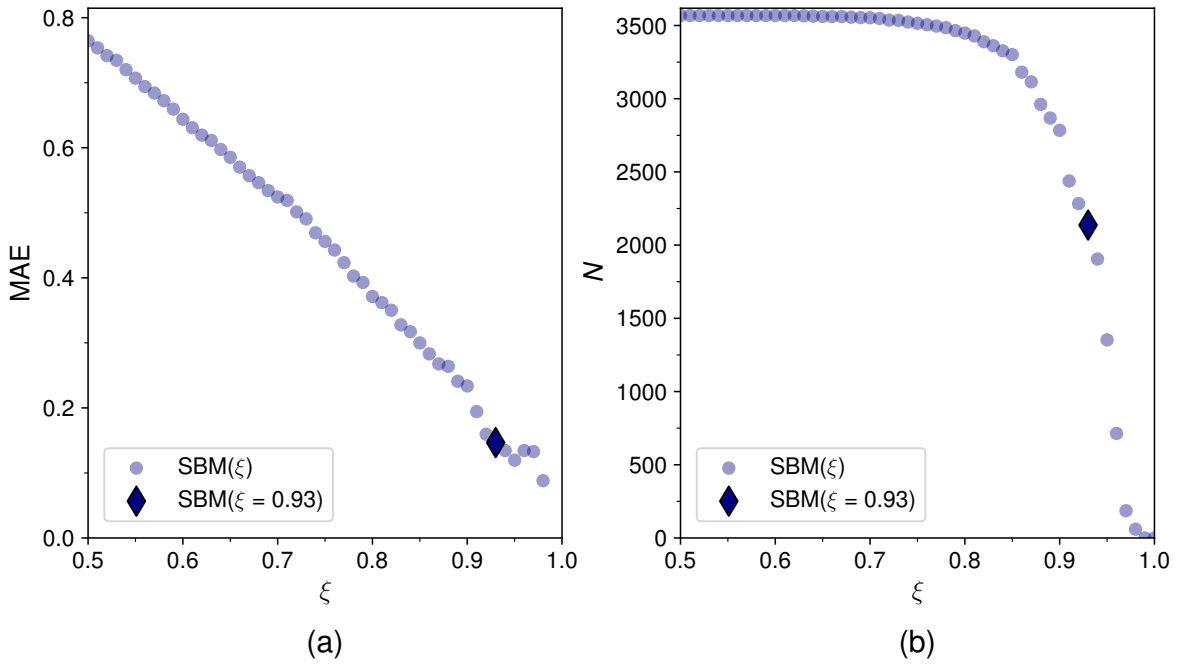


Figure 5: Mean absolute error (MAE) for the prediction of $\ln \gamma_{ij}^{\infty}$ (panel a) and number of predictable experimental data points N (panel b) of the best-performing SBM over the threshold ξ. The results for the SBM with ξ = 0.93 are highlighted.

Fig. 5a shows that increasing ξ results in a nearly linear decrease in MAE, indicating improving accuracy. In contrast, the relationship between N and ξ in Fig. 5b is more complex. For $\xi \leq 0.62$, the model achieves its maximum scope, i.e., predicting all experimental data points, while for $\xi > 0.98$, none of the mixtures can be predicted. Between these two boundaries, N first decreases slowly with increasing ξ, followed by a steep decrease as ξ approaches 1. This sensitivity of N to ξ emphasizes the importance

of selecting an optimal threshold. Overall, Fig. 5 supports the choice of $\xi = 0.93$, marked by the diamond, as a balance point that combines high predictive accuracy with substantial scope. While selecting a lower threshold would yield a broader scope, $\xi = 0.93$ is preferred here as it achieves an MAE in the range of typical experimental uncertainties.

A detailed analysis of the results for the similarity S_{mn} of all pairs of solutes and all pairs of solvents is presented in Fig. 6. The results are plotted in matrices, which are symmetric as $S_{mn} = S_{nm}$. In these matrices, the solutes (solvents) were arranged so that similar solutes (solvents) were positioned nearby, which was done using a clustering algorithm adopted from Ref. [57]. The chosen arrangement of the solutes (solvents) leads to high values of S_{mn} along the diagonal, cf. Fig. 6.

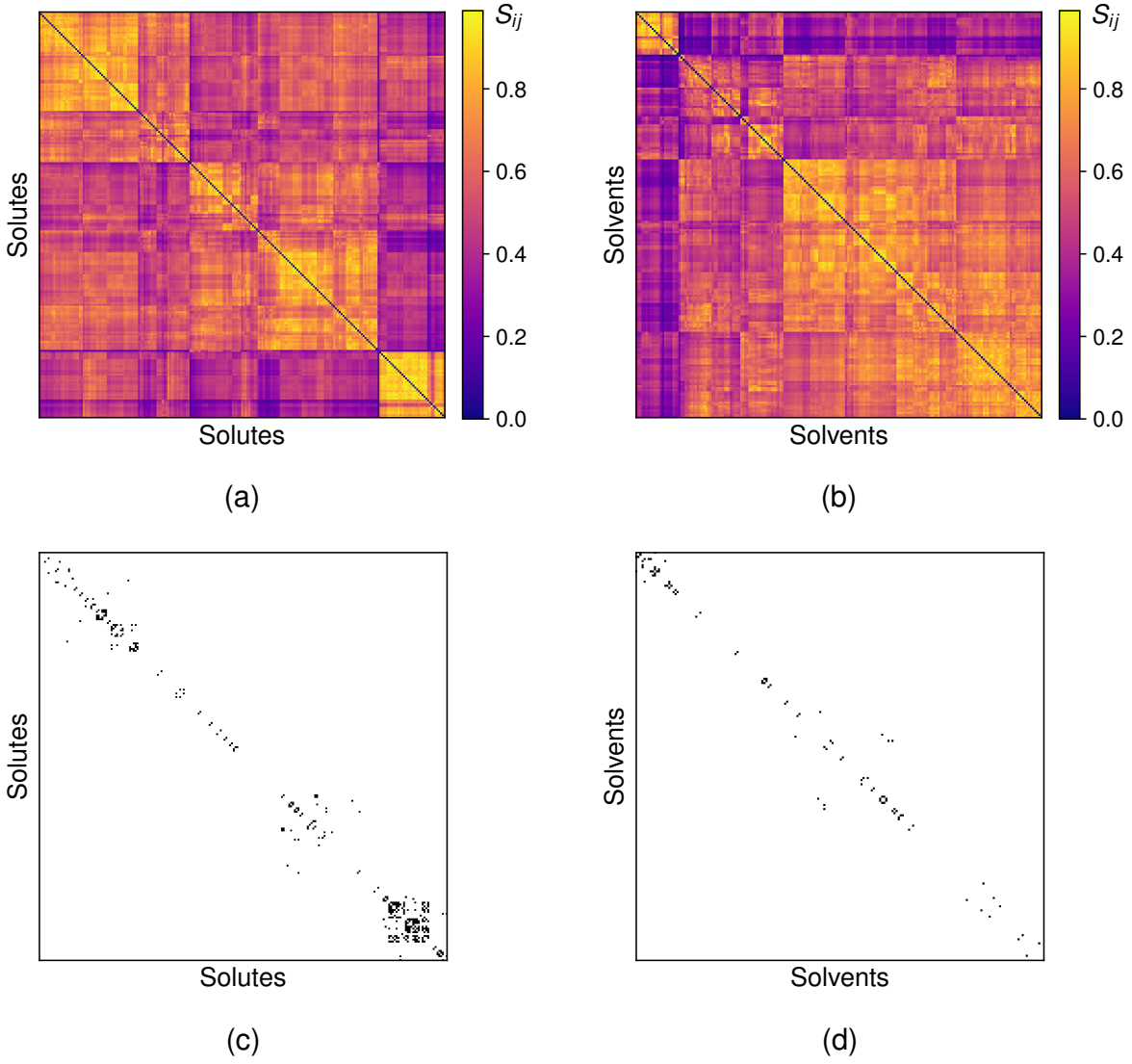


Figure 6: Heatmaps showing results for the pairwise similarity scores S_{mn} of the considered solutes (panel a) and solvents (panel b). For illustration, pairs with $S_{mn} > 0.93$ are highlighted in panels c and d.

The heatmaps in Fig. 6a and b reveal only a few strong similarities among the solutes and solvents in the database, as indicated by the few bright yellow areas. A notable exception is observed in the lower right corner of the solute matrix, cf. Fig. 6a, where a yellow square primarily represents alkanes classified as very similar according to the developed metric.

This observation becomes even more apparent when highlighting the solute-solute and solvent-solvent combinations with S_{mn} higher than $\xi = 0.93$, the threshold chosen for the detailed analysis discussed above, cf. Fig. 6c and d. Interestingly, only very few, or even just one, similar solutes or solvents for the mixture of interest are needed for the SBM to achieve the excellent predictive accuracy discussed earlier. Thus, for a set of similar mixtures, i.e., those with at least one similar solute or solvent according to the similarity metric, it is sufficient to measure $\ln \gamma_{ij}^{\infty}$ for just one of them. The other mixtures can then be predicted with high accuracy using the SBM. This finding is exciting for the planning of experiments in several ways. For example, it opens up ways to replace substances that are difficult to handle experimentally by suitable proxies, and it can also be used to devise strategies for an efficient design of experiments (DOE) to improve the accuracy and scope of the SBM with a minimum amount of additional experimental data.

3.5 Conclusions

This chapter has two primary outcomes: the first is a new way to measure the similarity between two components. It only needs the components' σ-profiles and their surface areas as input, information that can be obtained for basically any molecule from a quantum-chemical calculation or databases. Hence, the new measure of similarity is highly versatile. The information on these two properties of the two components m and n is compared and the result is combined in a similarity score S_{mn}, which is defined in such a way that the values always range between 0 and 1. The definition of this score contains hyperparameters (weights) that can be adapted to different tasks. In the present chapter, the goal was to use S_{mn} to develop a new method for predicting the limiting activity coefficient γ_{ij}^{∞} of a solute i in a solvent j in systems for which no experimental data are available. The hyperparameters have been chosen so that the resulting similarity scores are beneficial for this task. However, the resulting definition of the similarity score should also be helpful for many other tasks related to predicting or assessing the thermodynamic properties of binary liquid systems.

The second outcome of this chapter is the new similarity-based method (SBM) for predicting isothermal activity coefficients at infinite dilution γ_{ij}^{∞}. The idea behind the

method is simple. Starting with a database on γ_{ij}^{∞} at the temperature of interest, the value for a certain combination $i + j$ for which no data are available is predicted. Let us assume we have a data point for γ_{in}^{∞} for a system with the solute i of interest in combination with another solvent n. Whether the solvents j and n are sufficiently similar (i.e., $S_{jn} > \xi$) is simply checked and then the result for γ_{in}^{∞} is taken as a proxy for γ_{ij}^{∞}. (The same works if the problem is not the solvent but the solute). Of course, there must be rules on handling cases in which several such proxies are found. A simple arithmetic average is applied in this case, but taking the arithmetic average of the results of sufficiently similar substances is only one option; others could be explored. In the procedure of predicting γ_{ij}^{∞}, another hyperparameter is introduced, the threshold ξ. This threshold is left open, and the user can specify it. The functions that relate the chosen value of ξ to the number of systems for which predictions are possible and the expected accuracy of the prediction (measured, e.g., by the MAE obtained in a leave-one-out study) can be easily established, and give guidance for the application of the method. In general, the higher ξ, the more accurate the prediction will be, but high values of ξ will compromise the method's applicability.

The SBM that has been developed here for predicting γ_{ij}^{∞} shows a remarkable accuracy, even though the database is not large and typically contains only very few (if any) highly similar systems for any given combination of solute i and solvent j. The new SBM outperforms the established physical benchmark methods modified UNIFAC (Dortmund), COSMO-SAC, and COSMO-SAC-dsp.

The approach for designing SBMs based on the new similarity score S_{mn} is generic and can be transferred to any physical property of binary liquid mixtures. For thermodynamic applications, the hyperparameters of S_{mn} determined in the present chapter should be a good starting point but could be adapted for other applications.

The observation that data for only a few similar mixtures are sufficient to achieve accurate predictions suggests that a comparatively low number of targeted experiments can considerably improve SBMs. More generally, this finding could form the basis for new guiding principles for the design of experiments in binary systems.

4 Matrix Factorization of Thermodynamic Properties

4.1 Henry's Law Constants at 298 K

4.1.1 Introduction

Knowledge on the solubility of gases in solvents is essential for the design of many technical processes, such as gas absorption; and it is also needed for understanding many processes in nature. Gas solubility is usually described by Henry's law (cf. Eq. (B.1) in Appendix B), in which the key property is the Henry's law constant H_{ij}. The number of H_{ij} depends only on the temperature and the nature of the solute i and the solvent j. The solute is typically supercritical at the studied temperature, which is why it is called "gas". A large Henry's law constant H_{ij} corresponds to a low solubility and vice versa.

Experimental data on H_{ij} are scarce compared to the variety of possible combinations of relevant solutes and solvents. In the present chapter, new prediction methods for H_{ij} from the field of *machine learning* (ML) are introduced: *matrix completion methods* (MCMs). Various types of MCMs have been proposed in the literature [58–60], in particular for recommender systems [4, 5], and received a lot of attention through the Netflix Prize [6], an open competition of Netflix aiming at improving their system for the prediction of user rating for movies and TV shows. This chapter introduces MCMs for the prediction of H_{ij} at constant temperature in binary systems and thereby a Bayesian approach [8, 9, 61] is followed, which is known to be robust to overfitting without requiring much parameter tuning [62].

MCMs are highly interesting for predicting thermodynamic properties of binary systems. The idea behind this is that data for a given property of a binary system, such as the Henry's law constant H_{ij} of a solute i in a pure solvent j at a given temperature, can be stored conveniently in a matrix. The respective matrices containing the experimental data are typically very sparse, since the measurement of fluid properties is in general

tedious and expensive and, in addition, the number of components and systems of interest is large. The prediction of the missing entries in such a matrix constitutes a matrix completion problem. Refs. [8, 9] have recently introduced MCMs for the prediction of activity coefficients at infinite dilution in binary systems at constant temperature. In these studies, an in-depth discussion of the basic idea of applying MCMs for the prediction of thermodynamic mixture data is given. Here, this approach is extended to the prediction of H_{ij}.

In the present chapter, only pure solvents are considered and the temperature is fixed to 298.15 K±1 K (labeled as 298 K here, for simplicity), such that H_{ij} is fully specified by specifying the components i and j. The temperature dependence of the Henry's law constant is highly interesting, but was excluded from the present chapter, which is focused on introducing new methods for predicting H_{ij}. However, these methods can be extended to include the temperature dependence of properties once they are established for the isothermal case. A possible approach to implement such an extension for the prediction of activity coefficients at infinite dilution γ_{ij}^{∞} has been shown in Ref. [10], where the dependence of γ_{ij}^{∞} on the temperature T has been modeled by exploiting the fact that it can be well described by $\ln \gamma_{ij}^{\infty}(T) = A_{ij} + B_{ij}/T$ with system-specific, but temperature-independent, parameters A_{ij} and B_{ij} in many cases.

The accurate measurement of H_{ij} requires an extrapolation to the limiting case $x_i \to 0$, for which a series of experiments is necessary that makes these studies tedious. Therefore, experimental data on H_{ij} are missing for many practically relevant systems. This is why methods for predicting the Henry's law constant are so interesting.

Nevertheless, there are only comparatively few methods for predicting Henry's law constants so far. Most of these methods relate the Henry's law constant to physical component descriptors, mostly phenomenological descriptors like critical properties [63], molecular descriptors like molecular masses and polarizability [64, 65], or SMILES representations [66]. These *Quantitative Structure Property Relationships* (QSPR) [67] are often based on nonlinear approaches like artificial neural networks or support vector machines. In some cases, techniques from machine learning have been used for descriptor selection, such as the *Replacement Method* [68, 69] and *Genetic Algorithm* techniques [68, 70]. All of these methods are restricted to a special class of systems: they either only consider aqueous solutions [64–66, 68, 69, 71] or can only be applied for the prediction of the Henry's law constant of a single solute in different ionic liquids [63, 70]. Since the scope of these methods is very restricted, they are not considered further here.

In contrast, *group-contribution equations-of-state* (GC EoS), from which the Henry's law constant can be determined by well established routes [72], have a wider applicability. In GC EoS the EoS is typically combined with a mixing rule that is based on a model of the excess Gibbs energy (G^{E}). Using a group-contribution G^{E}-model, such as UNIFAC [41]

or modified UNIFAC (Dortmund) [13], then results in a GC EoS, of which several have been proposed in the literature [72–74]. The EoS used in these approaches are often simple cubic EoS for which the G^{E}-mixing rules are known to give good results for a large variety of systems [72, 75].

The group-contribution concept enables predictions for systems for which no data are available. The prerequisite for carrying out this calculation is, however, that the group interaction parameters of the G^{E}-model are available. Parameter matrices including typical supercritical solutes, as they are encountered in gas solubility problems, have been established for GC EoS [76]; however, the parameter tables are still far from covering all cases of interest.

One particularly successful GC EoS, which has also been implemented in commercial process simulators, is the *Predictive Soave-Redlich-Kwong* (PSRK) EoS [76, 77]. The PSRK EoS (simply named PSRK in the following for brevity) is a combination of the cubic Soave-Redlich-Kwong EoS [78] with a mixing rule based on the original UNIFAC model [41]. The parameter tables for PSRK include many supercritical compounds. Specifically, the current public parameter table of the PSRK model distinguishes 81 main groups and comprises fitted pair-interaction parameters for 956 combinations of them [76]. Based on the reported parameters, a large number of components and systems can be modeled, and the PSRK model has also demonstrated to yield reliable predictions for many different systems [76, 79], although its predictive accuracy decreases for highly asymmetric systems [80]. However, note that there is still a substantial number of missing pair-interaction parameters of the PSRK model, namely for 2284 combinations of the present main groups, that have not been reported yet, which hampers its applicability. PSRK is used here as a physical reference model for assessing the performance of the novel prediction methods based on matrix completion. Furthermore, the physics-based PSRK is used in the development of a novel hybrid prediction method by combining it with a data-driven ML method as described in detail in the following sections.

The Henry's law constant $H_{ij}(T)$ can in principle also be determined from the pure component vapor pressure $p_i^{\mathrm{s}}(T)$ and the activity coefficient at infinite dilution $\gamma_{ij}^{\infty}(T)$ using information on the Poynting correction as well as on the pure component saturated vapor fugacity coefficient; for details, see Eq. (B.4) in Appendix B. However, determining $H_{ij}(T)$ in this way implies that the solute i is subcritical at the temperature T. In this case, typically Raoult's law would be used to describe the equilibrium condition of the component i, rather than using Henry's law, so that the Henry's law constant is not needed at all. Nevertheless, a substantial part of the experimental literature data on $H_{ij}(T)$ refers to this case. These data were included here, but it is emphasized that the

main area of application of $H_{ij}(T)$ is the description of the solubility of supercritical components i.

This chapter is organized as follows: the database for H_{ij} used in this chapter is first described. Two different MCMs for predicting H_{ij} are then introduced: one that is completely data-driven and another that constitutes a hybrid of a data-driven MCM and the physics-based PSRK. Subsequently, the results are presented and discussed.

4.1.2 Database

The experimental data on Henry's law constants H_{ij} of solutes i in solvents j at 298 K used in the present chapter were taken from the Dortmund Data Bank (DDB) [81]. 298 K was chosen since at this temperature, by far the most data points for H_{ij} are reported in the DDB, as shown in Fig. B.1 in Appendix B. The raw data on H_{ij} were preprocessed as described in the following. Data points that were labeled to be of poor quality in the DDB were excluded. Furthermore, only solutes and solvents for which data for at least two different binary systems were available were considered, as this is a prerequisite for the application of the leave-one-out analysis as described in detail below. Finally, for those binary systems for which multiple data points in the temperature range of 298.15±1 K were available, the arithmetic mean of H_{ij} was calculated and used. The resulting data set comprises I = 101 solutes and J = 247 solvents and can, hence, be represented in a $I \times J$ matrix, which is depicted in Fig. 7; information on the considered solutes and solvents is summarized in Tables B.1 and B.2 in Appendix B, respectively. This matrix has 24,947 elements, but only 2,661 of them are occupied with experimental data, corresponding to 10.7%. In Fig. 7, the systems for which experimental data are available are represented as colored entries with the color code indicating the corresponding numerical value of H_{ij}, whereas the systems for which no experimental data are available are represented as black entries. The natural logarithm of H_{ij}, i.e., $\ln H_{ij}$, is thereby used in Fig. 7 and throughout this chapter for scaling purposes.

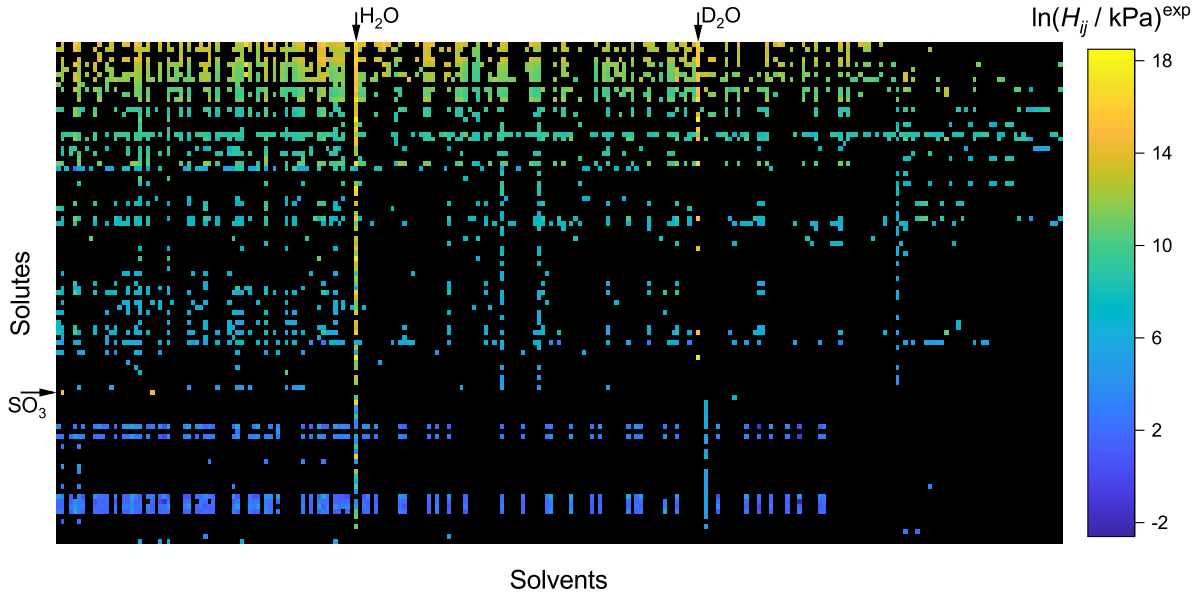


Figure 7: Matrix representing the experimental data on Henry's law constants H_{ij} of solutes i in pure solvents j at 298.15±1 K as reported in the DDB [81] after preprocessing (see text). The color code indicates the numerical value of $\ln H_{ij}$. The order of the solvents is arbitrary, while the solutes are arranged according to their critical temperature T_c according to the DDB from low (top) to high (bottom).

Only 16 of the 101 solutes are supercritical at the considered temperature. This is an extremely small number, considering the importance of gas solubility. In order to have a sufficiently large database, sub- and supercritical solutes were not differentiated in the present chapter and all available data in the DDB were simply operated on.

It is interesting that the entries in a single row in Fig. 7 show a fairly uniform color, i.e., for a given solute, the numbers of H_{ij} are similar for most solvents. In contrast, for a given solvent, the numbers of H_{ij} vary strongly, depending on the solute it is combined with. Furthermore, the color code indicating the values of H_{ij} in Fig. 7 reveals a strong correlation between the critical temperature of a solute and its solubility: for solutes with lower critical temperature, in general higher H_{ij} are observed and vice versa.

There are, however, a few apparent exceptions: most of the considered solutes are hydrophobic and therefore substantially poorer soluble in water (H_2O) and heavy water (D_2O) than in other solvents, cf. labeled columns in Fig. 7. Furthermore, the solute sulfur trioxide (SO_3) exhibits rather high Henry's law constants (poor solubilities) despite a comparatively high critical temperature; however, as for SO_3 only data for two solvents are available, this finding should not be overly interpreted.

Twenty-nine of the components are present both as solute and solvent in the data set, cf. Tables B.1 and B.2 in Appendix B. The corresponding solute-solvent combinations

would be pure components and were therefore not considered in the present chapter; for a detailed discussion of these cases, it is referred to Appendix B.

For subcritical solutes, Henry's law constants can also be calculated from the solute's vapor pressure and its activity coefficient at infinite dilution. In principle, this could have been used for augmenting the database on H_{ij} here. This option was considered but discarded, firstly, as it would have further increased the already large fraction of data for subcritical solutes, and, secondly, as the corresponding calculation requires assumptions on the fugacity coefficient of the solute and the Poynting correction, which introduce additional errors.

4.1.3 Matrix Completion Methods

Two different matrix completion methods (MCMs) were used in this chapter for predicting Henry's law constants H_{ij} for binary systems at 298 K. Both MCMs are based on a Bayesian approach [8, 9], which considers random variables drawn from a probability distribution instead of scalar parameters and which enables the incorporation of prior knowledge in a straightforward manner, as described in detail below. Both MCMs are collaborative-filtering methods [4, 27] that do not incorporate any direct information on the *pure components*, such as physical component descriptors, but use only the available *mixture data* for the binary systems, from which they infer so-called *latent variables* (LVs) during the training.

In both MCMs, the natural logarithm of H_{ij} is modeled as a stochastic function of LVs:

$$\ln H_{ij}^{\mathrm{MCM}} = \boldsymbol{u}_i \cdot \boldsymbol{v}_j + b_i^u + b_j^v \tag{9}$$

where $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$ are vectors of length K, whereas b_i^u and b_j^v are scalars. $\boldsymbol{u}_i$ and b_i^u represent the LVs of the solute i, $\boldsymbol{v}_j$ and b_j^v those of the solvent j. Hence, in both MCMs, each solute and each solvent is described by $K + 1$ component-specific LVs, which are determined from data on the mixture property $\ln H_{ij}$ (all LVs are initially unknown and inferred from the training data on $\ln H_{ij}$ during the training of the MCMs). K is a hyperparameter of the models and was set to $K = 4$ in all cases based on preliminary studies using cross-validation; however, the presented MCMs are robust regarding variations of K as demonstrated in Fig. B.10 in Appendix B.

The product $\boldsymbol{u}_i \cdot \boldsymbol{v}_j$ in Eq. (9) describes the contribution of specific pairwise interactions between solute i and solvent j to $\ln H_{ij}$, whereas b_i^u and b_j^v can be interpreted as a *general solubility* of a solute i and a *general dissolving capacity* of a solvent j, respectively, irrespective of specific binary interactions. In the following, b_i^u and b_j^v are referred to as *solute bias* and *solvent bias*, respectively, or summarized under the term *component*

biases. Such biases are also commonly considered for users and movies in recommender systems of, e.g., movie streaming services, where they take into account that some users are generally more critical than others when rating movies, and that some movies are in general rated higher than others [26]. They turn out to improve the model also in the present application for predicting H_{ij}. The consideration of the solute bias b_i^u is motivated in particular by the observation that some solutes show poor solubility in almost all studied solvents whereas other solutes are highly soluble in most solvents, cf. Fig. 7. A similar behavior was not observed for activity coefficients at infinite dilution, which was studied in Refs. [8, 9].

A Bayesian approach is used here, cf. Chapter 2. As inference method, variational inference [15, 19] was chosen. From the posterior, i.e., the inferred LVs, $\ln H_{ij}$ can also be predicted for previously unreported binary systems following Eq. (9). In each case, a probability distribution for $\ln H_{ij}$ is thereby predicted, which also provides information on the model uncertainties. In the following sections, the characteristics of the two MCMs developed in this chapter are discussed in more detail.

4.1.3.1 Data-Driven MCM

The first MCM is purely data-driven: its LVs are trained only to the sparse available experimental data for $\ln H_{ij}$ from the DDB, cf. Fig. 7; no other information is used. This method is referred to as *MCM-data* in the following. Fig. 8 shows an overview of how MCM-data is trained and used to predict $\ln H_{ij}$.

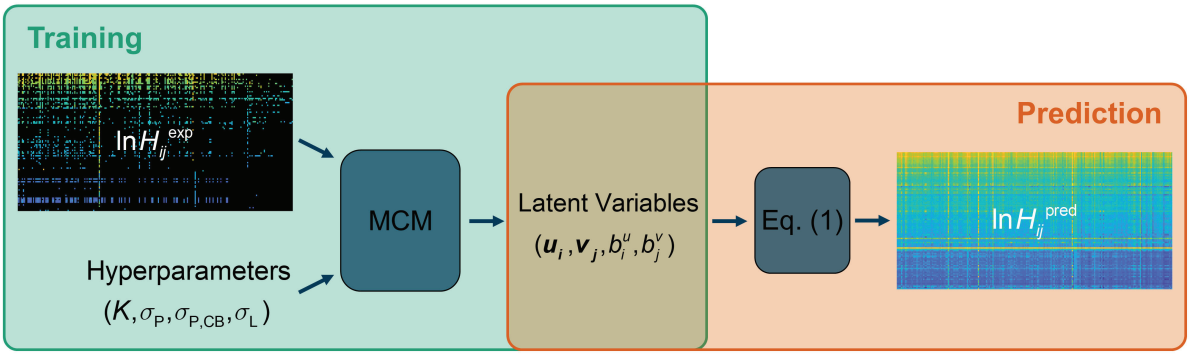


Figure 8: Schematic illustration of the prediction of $\ln H_{ij}$ with MCM-data. The MCM is trained on experimental data on $\ln H_{ij}$ (exp) with specified hyperparameters. The inferred LVs are subsequently used with Eq. (9) to obtain predictions (pred) for all possible solute-solvent combinations.

In the case of MCM-data, no information about the LVs is available prior to the training. Therefore, a rather broad, thus non-informative, probability distribution was used as prior here. Specifically, a normal distribution centered around zero and standard deviation $\sigma_P = 1$ for $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$, and $\sigma_{P,CB} = 10$ for b_i^u and b_j^v was chosen:

$$p(u_{i,k}) = \mathcal{N}(0,\ \sigma_{\mathrm{P}}),\ \text{for}\ k = 1...K \tag{10}$$

$$p(v_{j,k}) = \mathcal{N}(0,\ \sigma_{\mathrm{P}}),\ \text{for}\ k = 1...K \tag{11}$$

$$p(b_i^u) = \mathcal{N}(0,\ \sigma_{\mathrm{P,CB}}) \tag{12}$$

$$p(b_j^v) = \mathcal{N}(0,\ \sigma_{\mathrm{P,CB}}) \tag{13}$$

In general, the smaller the values for the standard deviations (σ_{P} and $\sigma_{\mathrm{P,CB}}$) are chosen, the stronger the LVs are restricted and the smaller is the influence of the training data on the LVs. In contrast, a very broad prior distribution (large values of σ_{P} and $\sigma_{\mathrm{P,CB}}$) barely constrains the LVs, such that the posterior is predominantly determined by the experimental data. The influence of the choice of the hyperparameters was investigated in preliminary studies. The values reported here represent good compromises between the extremes "too narrow, i.e., too restrictive" and "too broad, i.e., too irrelevant". However, the window, in which good results are obtained is wide and similar results as the ones presented below can also be obtained with other choices of the hyperparameters.

As likelihood, which models the probability of the data on $\ln H_{ij}$ conditioned on the LVs, a normal distribution with standard deviation $\sigma_{\mathrm{L}} = 0.2$ centered around $\boldsymbol{u}_i \cdot \boldsymbol{v}_j + b_i^u + b_j^v$ was chosen:

$$\begin{aligned} p(\ln H_{ij} \,|\, \boldsymbol{u}_i, \boldsymbol{v}_j, b_i^u, b_j^v) &= \mathcal{N}(\boldsymbol{u}_i \cdot \boldsymbol{v}_j + b_i^u + b_j^v,\ \sigma_{\mathrm{L}}) \\ &= \mathcal{N}(u_{i,1} \cdot v_{j,1} + ... + u_{i,K} \cdot v_{j,K} + b_i^u + b_j^v,\ \sigma_{\mathrm{L}}) \end{aligned} \tag{14}$$

The choice of the hyperparameter $\sigma_{\mathrm{L}} = 0.2$ was motivated by the uncertainty of the available experimental data, i.e., twice the value found on average for the experimental uncertainty was chosen. All data points were thereby treated equally, i.e., $\sigma_{\mathrm{L}} = 0.2$ was used throughout.

4.1.3.2 Hybrid MCM

The second MCM is hybrid, as it is not only trained on (sparse) experimental data on $\ln H_{ij}$ but also incorporates information from the Predictive Soave-Redlich-Kwong (PSRK) equation-of-state [76] in the form of PSRK predictions. Consequently, this MCM is referred to as *MCM-hybrid* in the following. MCM-hybrid is based on the so-called *Whisky approach* proposed for the prediction of activity coefficients at infinite dilution in Ref. [8] and is therefore only briefly discussed here; it is referred to Ref. [8] for more details. Fig. 9 shows an overview of how MCM-hybrid is trained and used to predict $\ln H_{ij}$.

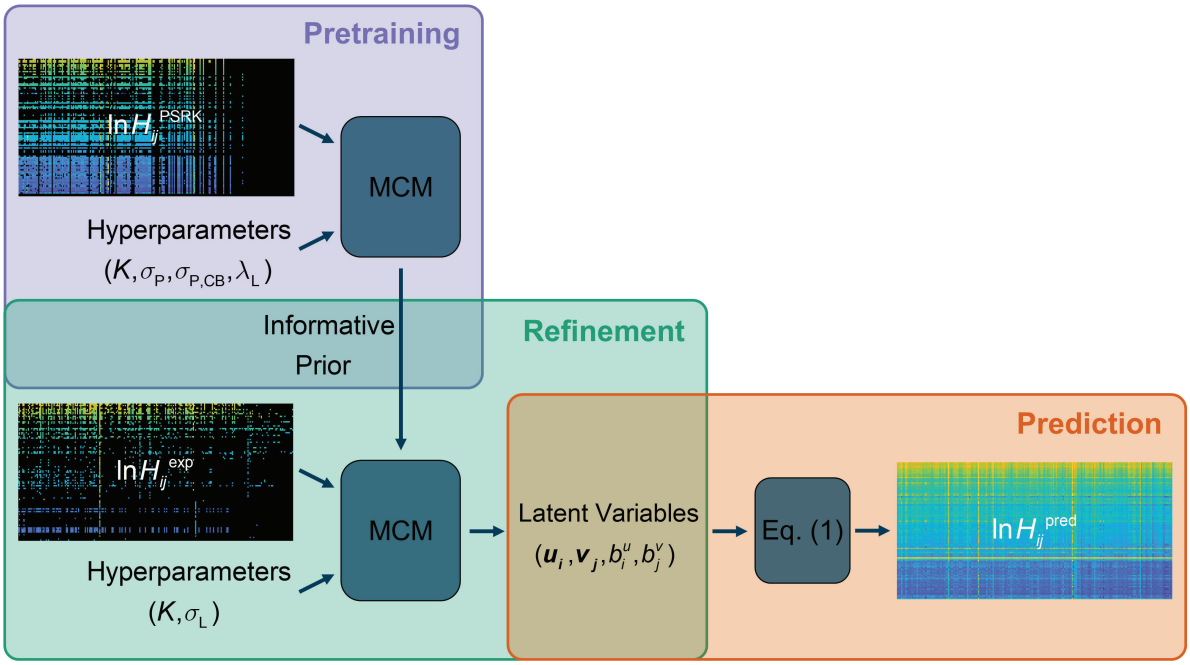


Figure 9: Schematic illustration of the prediction of $\ln H_{ij}$ with MCM-hybrid. In the pretraining step, the hyperparameters are specified and the MCM is trained on simulated data for $\ln H_{ij}$ from PSRK. The inferred (preliminary) LVs are used to generate an informative prior for the refinement step, in which the MCM is trained on experimental data on $\ln H_{ij}$ (exp). The resulting (final) LVs are subsequently used with Eq. (9) to obtain predictions (pred) for all possible solute-solvent combinations.

As MCM-data, MCM-hybrid models $\ln H_{ij}$ according to Eq. (9). However, in contrast to MCM-data, MCM-hybrid takes full advantage of the Bayesian approach to matrix completion by using an *informative* prior. The training of MCM-hybrid consists of two steps. In the first step, MCM-hybrid was trained on *simulated* data for $\ln H_{ij}$ that were generated with PSRK. With PSRK in its current public parameterization [76], predictions for 7,760 (31.1%) of all possible binary systems of the considered solutes and solvents can be obtained; hence, the matrix with this simulated data for the first training step is more densely occupied than the matrix with the experimental data, cf. Fig. 7. During the first training step, the MCM infers (provisional) LVs of the solutes and solvents from the predictions of PSRK, cf. Eq. (9). This step can be considered as *extracting* the physical knowledge on the solutes and solvents that is implicitly encoded in PSRK and explicitly provided in the form of PSRK predictions for $\ln H_{ij}$, and storing this knowledge in LVs. However, as the PSRK predictions are less reliable than the experimental data, the LVs obtained in this *pretraining* step are only preliminary and are therefore not directly used for predicting $\ln H_{ij}$. Instead, they are used to generate an *informative* prior for a second training step of the MCM. In the second training step, MCM-hybrid is, similarly to MCM-data, trained on the sparse set of available

experimental data on $\ln H_{ij}$. The second step can be understood as a revision of the preliminary LVs (inferred from the PSRK predictions alone) based on the experimental data; this step is referred to as *refinement* step in the following. The refinement step of MCM-hybrid yields the final set of LVs that contain information from the PSRK predictions *and* the experimental data. In the pretraining step of MCM-hybrid, the same broad normal distribution as in MCM-data, i.e., a normal distribution centered around zero with standard deviation $\sigma_{\mathrm{P,CB}} = 10$ for the component biases and $\sigma_{\mathrm{P}} = 1$ for the remaining LVs, was used as prior:

$$p(u_{i,k}) = \mathcal{N}(0,\, \sigma_{\mathrm{P}}),\ \text{for } k = 1...K \tag{15}$$

$$p(v_{j,k}) = \mathcal{N}(0,\, \sigma_{\mathrm{P}}),\ \text{for } k = 1...K \tag{16}$$

$$p(b_i^u) = \mathcal{N}(0,\, \sigma_{\mathrm{P,CB}}) \tag{17}$$

$$p(b_j^v) = \mathcal{N}(0,\, \sigma_{\mathrm{P,CB}}) \tag{18}$$

A Cauchy distribution with scale $\lambda_{\mathrm{L}} = 0.2$ was chosen as likelihood, which is in contrast to the training of MCM-data:

$$\begin{aligned} p(\ln H_{ij} \,|\, \boldsymbol{u}_i, \boldsymbol{v}_j, b_i^u, b_j^v) &= \mathrm{Cauchy}(\boldsymbol{u}_i \cdot \boldsymbol{v}_j + b_i^u + b_j^v,\, \lambda_{\mathrm{L}}) \\ &= \mathrm{Cauchy}(u_{i,1} \cdot v_{j,1} + ... + u_{i,K} \cdot v_{j,K} + b_i^u + b_j^v,\, \lambda_{\mathrm{L}}) \end{aligned} \tag{19}$$

The reason for using a Cauchy likelihood is that for some combinations of solutes and solvents, PSRK gives extremely (and unreasonably) large/small predictions for $\ln H_{ij}$ as shown in Fig. B.5 in Appendix B. These extreme outliers are attributed to badly chosen binary interaction parameters of PSRK; the problematic predictions are basically limited to hydrochloric acid (HCl) dissolved in alcohols. To prevent a negative impact due to these obvious outliers in the pretraining step of MCM-hybrid, the Cauchy distribution was chosen as it is more robust towards extreme outliers than the normal distribution.

Of course, the pretraining step can extract information from the PSRK predictions only for those solutes and solvents that can in general be modeled by PSRK, i.e., for which at least one $\ln H_{ij}$ within the considered matrix can be predicted. With the present public version of PSRK [76], this is the case for 81 of the 101 studied solutes (80.2%) and 142 of the 247 studied solvents (57.5%). Hence, only for those 81 solutes and 142 solvents, meaningful preliminary LVs can be inferred from the PSRK predictions and, as a consequence, an informative prior for the subsequent refinement step can be generated. For those solutes and solvents that can *not* be modeled by PSRK, the same uninformative prior as for the training of MCM-data was chosen in the refinement step

of MCM-hybrid: a normal distribution centered around zero with standard deviation $\sigma_{\mathrm{P}} = 1$ for $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$, and $\sigma_{\mathrm{P,CB}} = 10$ for b_i^u and b_j^v.

For those solutes and solvents that can be modeled by PSRK, an informative prior for the LVs in the refinement step was generated from the posterior of the pretraining step as described in the following. Since the posterior of the pretraining step of the studied LVs was approximately normally distributed in all cases, they were fitted with normal distributions, yielding a mean and standard deviation for each LV. The means were adopted, whereas the standard deviations of all informed solute and solvent biases were subsequently scaled with a constant factor, such that the mean of all resulting standard deviations was $\sigma_{\mathrm{P,CB}} = 5$; similarly, the standard deviations of the remaining informed LVs were scaled to yield a mean standard deviation of $\sigma_{\mathrm{P}} = 0.5$. The scaling factors, which can be seen as hyperparameters, were set to 6.44 for σ_{P} and 172.08 for $\sigma_{\mathrm{P,CB}}$, respectively, to obtain the specified mean standard deviations. This scaling procedure is necessary, since the predictions of PSRK are in general less trustworthy than the experimental data, and show some extreme outliers as exemplified in Fig. B.5 in Appendix B. Without this scaling, the predictions of PSRK and the experimental data would be basically treated in the same way, resulting in an exaggerated influence of the PSRK predictions on the training of the hybrid MCM. By setting the mean standard deviation to half of the values for the uninformed prior ($\sigma_{\mathrm{P}} = 1$ or $\sigma_{\mathrm{P,CB}} = 10$, cf. above), a stronger prior was obtained for those LVs for which a priori information could be extracted from the PSRK predictions. However, these informed prior probability distributions for the LVs are still broad enough to give enough flexibility in the refinement step, if sufficient evidence is provided by the experimental training data.

The scaling of the posterior distributions from the pretraining step can in general lead to distributions that are broader than the uninformed prior. Therefore, a last processing step was introduced to ensure that the informed prior for those solutes and solvents for which a priori information could be extracted from the PSRK predictions is *always* stronger than the uninformed prior for those solutes and solvents for which this is not the case. This was achieved by multiplying the scaled posterior from the pretraining step with the respective uninformative prior distributions, resulting in the final informative prior for the refinement step of MCM-hybrid:

$$p(u_{i,k}) = \mathcal{N}(u_{i,k}^*,\, \sigma_{\mathrm{P}}^*), \text{ for } k = 1...K \tag{20}$$

$$p(v_{j,k}) = \mathcal{N}(v_{j,k}^*,\, \sigma_{\mathrm{P}}^*), \text{ for } k = 1...K \tag{21}$$

$$p(b_i^u) = \mathcal{N}(b_i^{u*},\, \sigma_{\mathrm{P,CB}}^*) \tag{22}$$

$$p(b_j^v) = \mathcal{N}(b_j^{v*},\, \sigma_{\mathrm{P,CB}}^*) \tag{23}$$

Again, normal prior distributions were used for all LVs, but not centered around zero (as in the pretraining step of MCM-hybrid and the training step of MCM-data) but centered around an initial guess for each LV ($\boldsymbol{u}_i^*, \boldsymbol{v}_j^*, b_i^{u*}, b_j^{v*}$) based on the posterior of the preceding pretraining step; also the standard deviations of the prior distributions ($\sigma_{\mathrm{P}}^*, \sigma_{\mathrm{P,CB}}^*$) in the refinement step were set based on the posterior of the preceding pretraining step.

The final prior (informative for components that can be modeled with PSRK, uninformative for components that can not be modeled with PSRK) was ultimately used in the refinement step of MCM-hybrid, in which the method was trained on the available experimental data for $\ln H_{ij}$ from the DDB, cf. Fig. 7. In the refinement step, a normal likelihood with standard deviation $\sigma_{\mathrm{L}} = 0.2$ was chosen, which is in analogy to the (single) training step of MCM-data:

$$\begin{aligned} p(\ln H_{ij} \,|\, \boldsymbol{u}_i, \boldsymbol{v}_j, b_i^u, b_j^v) &= \mathcal{N}(\boldsymbol{u}_i \cdot \boldsymbol{v}_j + b_i^u + b_j^v,\ \sigma_{\mathrm{L}}) \\ &= \mathcal{N}(u_{i,1} \cdot v_{j,1} + \ldots + u_{i,K} \cdot v_{j,K} + b_i^u + b_j^v,\ \sigma_{\mathrm{L}}) \end{aligned} \tag{24}$$

Similar to MCM-data, preliminary studies have shown that MCM-hybrid exhibits robust behavior for hyperparameters over a wide range. The procedure can be adapted as needed, but the one proposed here works well for predicting Henry's law constants.

Computational Details

Both MCMs introduced in this chapter were implemented in the probabilistic programming language *Stan* [20]; details on the models including the source code to run them in Stan are given in Figs. B.2 - B.4 in Appendix B. As inference method, which inverts the generative process of the probabilistic model and reasons about the LVs for given data ($\ln H_{ij}$ here), Gaussian mean-field variational inference [19] was used. All pre- and postprocessing steps were performed in MATLAB® R2019b [21].

For evaluating the predictive performance of the two MCMs, a leave-one-out analysis [56] was used. Each MCM was thereby trained multiple times, and in each run, one of the 2,661 available experimental data points was withheld during the training and subsequently predicted by the MCM; the prediction was then compared to the withheld experimental value. This leave-one-out analysis requires that for each solute and each solvent at least two data points for different systems are available. If this condition is satisfied, there is at least one data point for each component in the training set (after withholding the test data point), such that the MCMs have at least some information for learning the features of each component. Considering the deviations between prediction and experimental value of all available data points, the overall scores *mean absolute error* (MAE) and *mean squared error* (MSE) were calculated and compared among the

MCMs and with those from PSRK. The latter comparison is, however, not trivial: both MCMs developed here allow for the prediction of $\ln H_{ij}$ for *all possible* combinations of the studied solutes and solvents and therefore for notably more binary systems within the considered matrix than PSRK. They are assessed using leave-one-out analysis, i.e., based on real predictions, since the respective data point was not used for training the MCMs. In contrast, the deviations reported for PSRK are simply those from the trained method as it is reported in the literature [76]. Unfortunately, the training set that was used for obtaining the parameters of PSRK has not been disclosed in the literature. It may be speculated that it contained a large fraction of the data points that are considered here. Hence, even though formally the same statistical quantities are used to characterize the deviations for the MCMs on one side and PSRK on the other, they refer to different types of deviations. Of course, in contrast to the MCMs, PSRK can as a group-contribution method be used to describe additional components besides the 101 solutes and 247 solvents considered here.

In Ref. [82], all "final" LVs for all solvents and solutes are reported. They were inferred by MCM-hybrid using *all* 2,661 experimental data points for $\ln H_{ij}$ (without applying a leave-one-out strategy). The idea behind this is to obtain a single set of parameter values that enables a direct application of the MCM for predicting $\ln H_{ij}$. Comparing the numbers for the LVs reported in Ref. [82] and those obtained in the leave-one-out analysis reveals, as expected, only minor differences. Consequently, the numerical values in Ref. [82] constitute a *complete* parameter set of the *final* MCM-hybrid model and allows the prediction of $\ln H_{ij}$ for any binary combination of the studied solutes and solvents at 298 K.

4.1.4 Results and Discussion

In Fig. 10, the performance of the two developed matrix completion methods (MCM-data and MCM-hybrid) for the prediction of Henry's law constants $\ln H_{ij}$ in binary systems of a solute i and a solvent j at 298 K is evaluated in terms of MAE and MSE and compared to the performance of PSRK [76]. As described above, the scores of the MCMs are thereby obtained by applying a leave-one-out analysis and comparing the predictions with the respective experimental data from the Dortmund Data Bank (DDB) [81]. Note that in addition to the two MCMs discussed here, a third MCM was tested. It is a variant of MCM-data but without considering component biases, cf. Eq. (9). The results are presented in Appendix B and show that using the biases yields substantially better results.

In Fig. 10, only those data points that can be described with PSRK are considered. By using the latest published parameterization given by Horstmann et al. [76], PSRK can

predict $\ln H_{ij}$ for 1,438 of the 2,661 binary systems (54.0%) for which experimental data are available in the DDB [81], cf. Fig. 7.

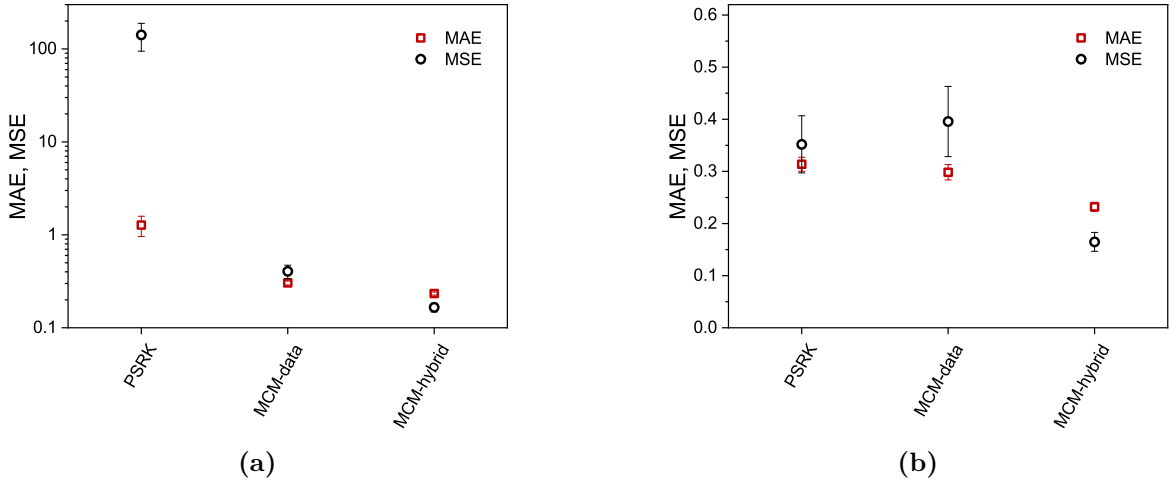
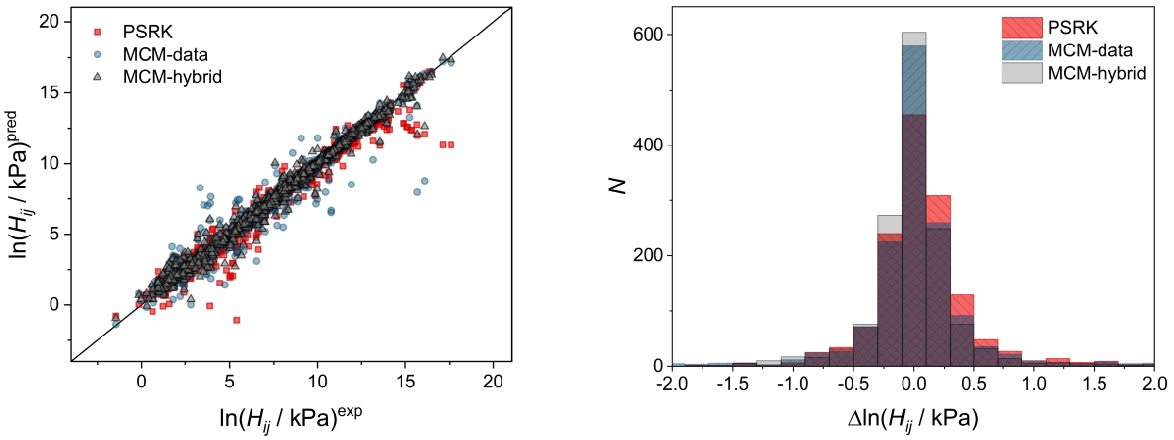


Figure 10: Mean absolute error (MAE) and mean squared error (MSE) of PSRK, MCM-data, and MCM-hybrid for the prediction of $\ln H_{ij}$ for binary systems at 298 K. (a) Considering the full data set (1,438 data points). (b) Without considering the worst 11 outliers of PSRK, cf. Fig. B.5 in Appendix B.

The results in Fig. 10a show that the MAE and MSE of PSRK are substantially larger than the respective scores of both MCMs (note the logarithmic scale). However, a closer analysis shows that the poor scores of PSRK can mainly be attributed to only a handful of data points that are extremely badly predicted by PSRK, as exemplified in Fig. B.5 in Appendix B. Most of these extreme outliers correspond to the solute hydrochloric acid (HCl) dissolved in alcohols as solvents and can be attributed to poor group-interaction parameters between the HCl group and the alcohol group of PSRK. To obtain a fairer comparison, these extreme outliers have also been omitted for calculating the MAE and MSE of the methods and the respective scores are represented in Fig. 10b.

When omitting the PSRK outliers, the performance of MCM-data is similar to that of PSRK. The hybrid approach MCM-hybrid clearly outperforms PSRK and MCM-data in both scores irrespective of whether the PSRK outliers are taken into account or not. It is interesting to realize that MCM-hybrid, which combines information from PSRK predictions with scarce experimental data in the training, apparently does not suffer from the extreme PSRK outliers. This underpins the robustness of MCM-hybrid. The results shown in Fig. 10 also demonstrate that the Bayesian approach for the hybridization works well and combines advantages of PSRK with those of the data-driven MCM, while not being impaired by the weaknesses of the individual methods.

In Fig. 11, the predictions of PSRK, MCM-data, and MCM-hybrid are compared in a parity plot (panel a) and a histogram representation of the deviations from the experimental data (panel b).



(a) (b)

Figure 11: Comparison of the predictions (pred) for $\ln H_{ij}$ with PSRK, MCM-data, and MCM-hybrid without considering the worst 11 outliers of PSRK. (a) Parity plot of predictions over experimental data (exp) from the DDB. (b) Histogram of the deviations of the predictions from the experimental data. N is the number of binary systems. The shown interval in the histogram contains 97.8% (PSRK), 98.4% (MCM-data), and 99.5% (MCM-hybrid) of all considered data points.

The representations in Fig. 11 support the findings described above. Fig. 11a clearly indicates that the hybrid approach particularly improves the prediction of those data points that are rather poorly predicted with PSRK or MCM-data (or both), which is consistent with the observation of a substantially lower MSE in Fig. 10. This again indicates that MCM-hybrid represents an extremely robust combination of two approaches that benefits from additional information but is not notably prone to shortcomings of the individual methods. Furthermore, Fig. 11b illustrates that MCM-hybrid predicts most data points with a very high accuracy; the deviations are often in the range of $|\Delta \ln (H_{ij}/\mathrm{kPa})| < 0.1$, corresponding to deviations that are in the order of the experimental uncertainty in the determination of Henry's law constants. For instance, the experimental uncertainty of $\ln H_{ij}$ has been estimated by calculating the mean standard deviation for those binary systems for which multiple data points in the temperature range of 298.15±1 K were available in the DDB and a value of almost exactly 0.1 was found.

Unlike the proposed MCMs, PSRK is, as group-contribution method, also able to model systems outside the considered matrix, which is not the case for all MCMs presented here. However, one major disadvantage of PSRK is that its application is limited to those

components and systems for which the method has been parameterized. As described above, only about 54.0% of the experimental data on H_{ij} taken from the DDB in this chapter can be predicted with the present public version of PSRK, which is why the comparison in Figs. 10 and 11 was only carried out based on those 54.0% of the data points. This restriction does not apply for the MCMs developed in this chapter, as they allow the prediction of H_{ij} of *all* possible binary systems of the considered solutes and solvents. This enables the evaluation of the predictive performance of the MCMs based on all 2,661 available experimental data points for H_{ij} by leave-one-out analysis, which is discussed in the following.

Fig. 12 shows the MAE and MSE of the predictions with MCM-data and MCM-hybrid; Fig. B.11 depicts the predictions of both methods in a parity plot (panel a) and a histogram representation (panel b) similar to Fig. 11. In Fig. B.12 in Appendix B, a parity plot that additionally includes information on the model uncertainties is given.

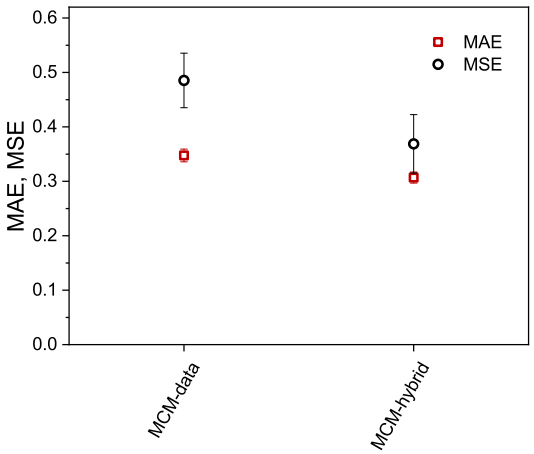
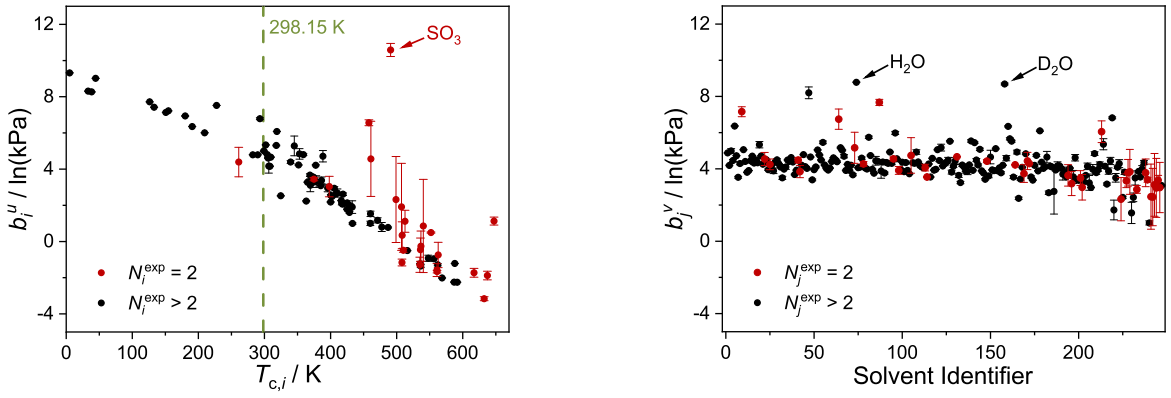


Figure 12: Mean absolute error (MAE) and mean squared error (MSE) of MCM-data and MCM-hybrid for the prediction of $\ln H_{ij}$ in binary systems at 298 K. For the evaluation, *all* 2,661 experimental data points from the DDB were considered here.

The observations are similar to those discussed above. The scores are slightly worse when all available data are considered than when only data that can be modeled with PSRK are considered. This is not unexpected, as those components that cannot be described with PSRK are in general less studied, i.e., for these components, less data for training the MCMs are available.

In the following, it is briefly discussed how MCMs can not only be applied for the prediction of mixture properties ($\ln H_{ij}$ here) but also enable interesting physical insights into the mixture data. Therefore, the LVs of the solutes and solvents inferred during the training of MCM-hybrid from the mixture data (PSRK predictions and experimental data on $\ln H_{ij}$) are studied in more detail.

Fig. 13 shows the component biases b_i^u and b_j^v of all solutes (panel a) and all solvents (panel b), respectively; the solutes and solvents are ordered in analogy to Fig. 7, i.e., the solutes are sorted according to their critical temperature in ascending order, while the solvents are arranged by their DDB number (which is rather arbitrary). Similar figures for the remaining LVs ($\boldsymbol{u}_i$ and $\boldsymbol{v}_j$) are shown in Figs B.13 and B.14 in Appendix B.



(a) **(b)**

Figure 13: Component biases of all solutes ((a), ordered according to the critical temperature) and solvents ((b), ordered according to the DDB number) as inferred by MCM-hybrid. Means (symbols) and standard deviations (error bars) were calculated from the results of the leave-one-out runs assuming normal distributions for the predictions. Solutes and solvents for which only data for two different systems are available in the data set are marked red.

The number of data points that was considered for each solute (solvent) in Fig. 13 equals the number of different binary systems in the data set that contain the respective solute N_i (solvent N_j). This number of data points is attributed to the performed leave-one-out analysis, where one experimental $\ln H_{ij}$ was withheld in each run and all LVs were trained. Thereby, only those LVs were saved that were obtained when the considered solute (solvent) was part of the one system that was withheld. From the selected data points, mean and standard deviation of b_i^u (b_j^v) were calculated and are depicted as symbols and error bars in Fig. 13, respectively. While $N_i = 2$ and $N_j = 2$ often lead to high standard deviations, rather small standard deviations are observed for most solutes and solvents that appear at least three times in the data set, i.e., for $N_i \geq 3$ and $N_j \geq 3$, respectively.

In Refs. [8, 9], where matrix completion methods were employed for the prediction of activity coefficients at infinite dilution, no component biases were used. This is motivated by the fact that there is no such thing as a solute that exhibits *in general* small (or *in general* large) activity coefficients in *any* solvent, and, analogously, there is no solvent that *in general* leads to small (large) activity coefficients of *any* solute. By contrast,

there are, e.g., gases whose solubility is *in general* rather small (or large) *irrespective* of the solvent, and this fact is taken into account by considering component biases for the prediction of Henry's law constants here; of course, a single gas does not exhibit the *exact same* solubility in all solvents, which is taken into account by the other latent variables that are considered. For the solute bias b_i^u, a clear correlation with the solute's critical temperature $T_{c,i}$ is found: b_i^u decreases with increasing $T_{c,i}$, cf., Fig. 13a. This is consistent with Fig. 7 and the expectation that solutes with high critical temperatures generally have a higher solubility than solutes with low critical temperature. For instance, helium and hydrogen, which have a very low critical temperature, are quite poorly soluble irrespective of the considered solvent. For the solvent bias b_j^v, no trend and only rather small variations are found (except for "extreme" molecules like water and heavy water), cf. Fig. 13b, which supports the hypothesis discussed in the analysis of Fig. 7 that the type of solute has a stronger influence on H_{ij} than the type of solvent. These observations do not only allow interesting physical insights, but also open the path for an estimation of the solute and solvent biases of components that are not included in the current data set. For instance, b_i^u could roughly be estimated from $T_{c,i}$ using the correlation shown in Fig. 13a, whereas for b_j^v, the average value of all solvent biases depicted in Fig. 13b could be used. The situation is more complicated when the other LVs are considered, cf. Figs. B.13 and B.14 in Appendix B. However, the examples shown in Fig. 13 underline that correlations of the LVs with physical descriptors can be found, even though they may not be as simple as in these fortunate cases.

4.1.5 Conclusions

In the present chapter, a new class of prediction methods for Henry's law constants H_{ij}, namely matrix completion methods (MCMs), has been introduced and their applicability for H_{ij} of solutes i in pure solvents j at 298 K has been demonstrated. The idea behind this approach is that binary data can conveniently be stored in a matrix and that MCMs, which are well established in machine learning, can be applied for completing matrices even in cases where they are only sparsely occupied with experimental data, as it is the case for H_{ij} (and many other mixture properties). Two MCMs for predicting H_{ij} were implemented in the present chapter using a Bayesian framework and the probabilistic programming language Stan. The first MCM is purely data-driven, i.e., it is trained only on the scarce available experimental data on H_{ij}, while the second MCM follows a hybrid approach by additionally incorporating predictions from the Predictive Soave-Redlich-Kwong (PSRK) equation-of-state. The performance of both MCMs for predicting H_{ij} for 101 solutes i and 247 solvents j was evaluated by a leave-one-out analysis using experimental data from the Dortmund Data Bank (DDB) [81]. While with the purely

data-driven MCM a predictive accuracy comparable to that of PSRK was found, a substantially better performance was obtained with the hybrid MCM.

The introduced MCMs have broad applicability: they are capable of predicting the H_{ij} for all 24,947 possible binary systems of the considered solutes and solvents as they are not limited by unavailable parameters; in contrast, PSRK can only predict $\ln H_{ij}$ for 31.1% of these binary systems. Of course, as group-contribution method, PSRK can in principle also be applied for predicting H_{ij} in systems containing other solutes and solvents than those studied here. Furthermore, while the refinement and extension of physics-based prediction methods like PSRK is very elaborate, the MCMs presented in this chapter can be adapted in a straightforward manner when additional data become available. Moreover, the presented matrix completion approach is not restricted to the prediction of Henry's law constants but can be transferred to other thermodynamic properties in a straightforward manner. The success of the MCMs is thereby based on uncovering structure in the respective mixture data, which can also be expected for many other thermodynamic properties.

4.2 Temperature-Dependent Henry's Law Constants

4.2.1 Introduction

Information on the gas solubility is of fundamental importance for designing and optimizing many chemical processes, such as gas absorption. Gas solubilities can be described by Henry's law constants H_{ij}, where i is the gas dissolved in the solvent j. Even if only binary systems, i.e., the solubility of a pure-component gas in a pure-component solvent, are considered, the number of relevant possible combinations exceeds those that have been studied in experiments by far, making predictive methods for H_{ij} indispensable in practice.

Group-contribution (GC) equations-of-state (EoS) have become a cornerstone for predicting gas solubilities by combining an EoS with a mixing rule based on a GC model of the excess Gibbs energy (G^{E}) [72]. However, the applicability of this approach depends on the availability of parameters for all structural groups in the system, and even for the most successful GC EoS, the Predictive Soave-Redlich-Kwong (PSRK) EoS [76, 77], significant gaps in the parameter table remain, preventing its application for many systems. PSRK serves as a baseline model, providing a benchmark against which the accuracy of the newly developed prediction methods are assessed.

A data-driven alternative to physical prediction methods for properties of binary systems is using matrix completion methods (MCMs) from machine learning, which are well established in recommender systems [4–6]. In recent work, it has been demonstrated that the MCM concept can be transferred to thermodynamics [1, 2, 8–10, 51, 54, 57, 82–85]. The basic principle of using MCMs in this field is to organize data of a given property of binary mixtures into matrices, where the matrix's rows and columns correspond to the components that make up the mixtures. These matrices are only sparsely populated with experimental data in basically all cases, as data are missing. However, from the available mixture data, similarities between the different components can be inferred and used to predict the missing entries, hence completing the matrix.

Specifically, an MCM was introduced in Chapter 4.1 for predicting H_{ij} for binary systems at a constant temperature of 298 K, where most data points for H_{ij} have been reported in the literature. While this MCM outperformed PSRK in prediction accuracy, it cannot generalize over the temperature, which hampers its applicability. Therefore, in the present chapter, the MCM from Chapter 4.1 is extended to a model for the prediction of temperature-dependent $H_{ij}(T)$.

The temperature-dependent $H_{ij}(T)$ data can be represented in a three-dimensional tensor, where the third dimension is used to include the temperature. Tensor completion

methods could, in principal, be used to fill this tensor [11]. However, in the present chapter, a strategy similar to recent advancements in predicting activity coefficients at infinite dilution γ_{ij}^{∞} [10] is adopted, and a physical equation is employed that captures the temperature dependence by introducing system-specific, yet temperature-invariant, parameters, which can be represented in matrices. These matrices are, again, only sparsely populated since experimental data for parameter fitting are only available for some systems. This way, the tensor completion problem is reduced to a problem of the completion of some matrices (here three) and an MCM for predicting their entries is developed.

This chapter is organized as follows: the database for $H_{ij}(T)$ used for training and evaluating the developed prediction methods is first described. Two different MCMs for predicting $H_{ij}(T)$, which are based on a physical three-parameter approach for modeling the dependence of H_{ij} on temperature, are then presented: one is fully data-driven, and the other is a hybrid of a data-driven MCM and the physics-based PSRK EoS. Subsequently, the results are presented and discussed.

4.2.2 Database

Experimental data on Henry's law constants $H_{ij}(T)$ in binary systems covering the temperature range between 173.15 K and 573.15 K were taken from the Dortmund Data Bank (DDB) [86], which is the largest database of thermodynamic mixture properties. The DDB compiles experimental data on gas solubility collected from various sources dating back over a century to the present day, making it an ideal basis for model training and validation. Only a few reported raw data points for $H_{ij}(T)$ at extreme temperatures were a priori excluded (921 out of 63,484), as depicted in Fig. C.1 in Appendix C.

The preprocessing of the data was conducted as follows. Since the experimental H_{ij} values span several orders of magnitude, $\ln H_{ij}$ was consistently applied for scaling purposes. Data labeled as poor quality in the DDB were discarded. Additionally, for binary systems with multiple data points within a 1 K temperature range, such as 298.15±0.5 K, the median of $\ln H_{ij}(T)$ was computed and utilized. Furthermore, a prerequisite for learning the temperature dependence of $\ln H_{ij}(T)$ is the availability of data points measured at various temperatures. Consequently, only systems with data spanning a temperature range of at least 10 K were considered.

For assessing predictive accuracy by leave-one-out analysis, data for at least two distinct binary systems per solute and solvent have to be available. Hence, all data sets that did not fulfill this requirement were discarded. The application of the different criteria resulted in a data set comprising 20,565 data points for 3,297 binary systems involving

I = 122 solutes and J = 399 solvents, as detailed in the Excel files supplied in Ref. [87]. In Fig. 14, all considered binary systems are displayed in an $I \times J$ matrix, using a color code representing the number of data points for each system (i.e., the number of different temperatures at which the system has been studied); black indicates the absence of data.

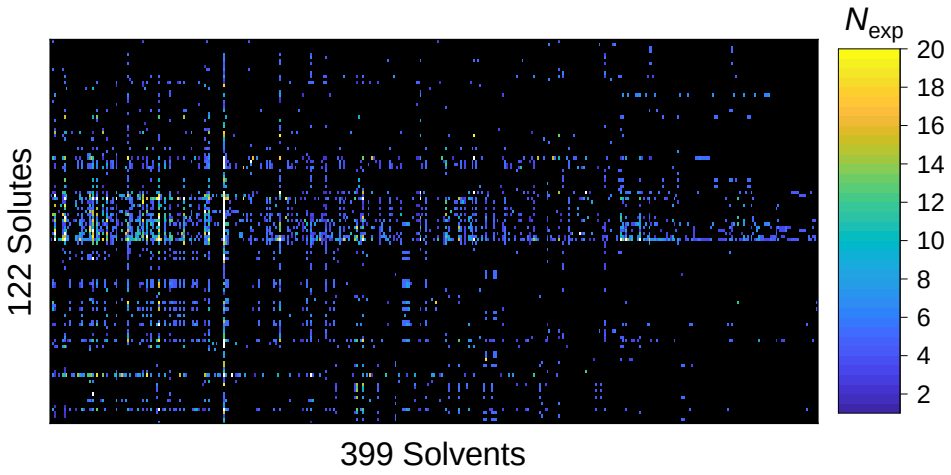


Figure 14: Matrix representing the number of experimental data points N_{exp} for $H_{ij}(T)$ of solutes i (rows) in solvents j (columns) between 173.15 K and 573.15 K as reported in the DDB [86] after preprocessing (see text). The order of the solutes and solvents is arbitrary. White color corresponds to binary systems studied at more than 20 different temperatures, black color indicates that no experimental data are available.

Fig. 14 shows that the experimental data cover only a small fraction (6.77%) of the potential 48,678 solute-solvent combinations in the matrix. Furthermore, the heatmap's color coding reveals that most systems have been studied at only a few different temperatures. Specifically, for 65% of the studied systems, no more than five data points are available. A notable exception is water, which has been subject to considerably more extensive research than any other solvent.

In the data set, 32 components are present both as solute and solvent. The corresponding solute-solvent combinations would result in pure components and have thus been excluded from this study.

4.2.3 Matrix Completion Methods

This chapter introduces two probabilistic MCMs to predict temperature-dependent Henry's law constants $H_{ij}(T)$. The first MCM is purely data-driven, relying exclusively on experimental data on $\ln H_{ij}(T)$. The second MCM is a hybrid approach incorporating predictions from the PSRK model as additional training data. Although

other equations-of-state are also possible choices for hybridization, PSRK is used because of its broad applicability, resulting from the large number of publicly available group-interaction parameters [76].

Both MCMs are only trained on the available *mixture data* for the binary systems (here $\ln H_{ij}(T)$), from which they infer component-specific characteristics, so-called *features*, during the training, without the need of any additional information on the pure components. In a Bayesian modeling approach, these features are considered to be random variables drawn from a probability distribution instead of scalar parameters, cf. Chapter 2.

The experimental data span a three-dimensional tensor based on solute i, solvent j, and (discretized) temperature T, necessitating reduction to two-dimensional matrices for applying MCMs. Therefore, the temperature dependence of $\ln H_{ij}(T)$ is modeled by [88]:

$$\ln(H_{ij} \,/\, \mathrm{kPa}) = A_{ij} + \frac{B_{ij}}{T \,/\, \mathrm{K}} + C_{ij} \left(\ln\left(\frac{T}{T_0}\right) + \frac{T_0 - T}{T} \right) \tag{25}$$

where the reference temperature is $T_0 = 298$ K. The three system-specific but temperature-independent parameters P_{ij} (A_{ij}, B_{ij}, C_{ij}) can each be arranged in a matrix spanned by the solutes i and solvents j, which is the prerequisite for using MCMs. Both MCMs developed in the present chapter predict each parameter P_{ij}^{pred} using a stochastic function of features:

$$P_{ij}^{\mathrm{pred}} = \boldsymbol{u}_i \cdot \boldsymbol{v}_j + b_i^u + b_j^v \quad \forall P \in (A, B, C) \tag{26}$$

where $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$ are vectors of length K, and b_i^u and b_j^v are scalars representing solute and solvent biases, respectively. This formulation leads to each solute and solvent being characterized by $K + 1$ features per parameter. As for the prediction of H_{ij} at 298 K (cf. Chapter 4.1), the predictive accuracy was very robust to variations in K in preliminary studies, based on which $K = 3$ was selected for both models.

The dot product $\boldsymbol{u}_i \cdot \boldsymbol{v}_j$ in Eq. (26) models the specific pairwise interactions between solute i and solvent j. In contrast, the component-specific biases b_i^u and b_j^v relate to the intrinsic contributions of the individual solutes and solvents, respectively. These biases are referred to here as *solute bias* for b_i^u and *solvent bias* for b_j^v, or, collectively, as *component biases*. It was demonstrated in Chapter 4.1 that using these component biases significantly improves the predictive accuracy of MCMs, and they have therefore been employed here as well.

In both MCMs introduced here, the features describing the parameter matrices A_{ij}, B_{ij}, and C_{ij} are not inferred individually but simultaneously through end-to-end training on

the $\ln H_{ij}$ data. To ensure consistency across the parameter matrices and facilitate this simultaneous learning process, scaling factors to the parameters B_{ij} and C_{ij} are applied: B_{ij} is scaled by 10^{-2} and C_{ij} by 10^{-1}. This adjustment results in values of A_{ij}, B_{ij}, and C_{ij} of similar order of magnitude.

To determine the posterior, Gaussian mean-field VI is employed through the *Automatic Differentiation Variational Inference* (ADVI) algorithm [19], implemented in the probabilistic programming language *Stan* [20]. The *MatlabStan* implementation was utilized, allowing the Stan code to be directly embedded in *MATLAB* [22] scripts. MATLAB was also used to perform all necessary pre- and postprocessing steps.

In this chapter, the predictive accuracy of the MCMs is evaluated using leave-one-out analysis [56]. Therefore, an MCM is trained on the available data, excluding those for one binary system, which are subsequently used as the test set. This approach enables the model to learn the characteristics (features) of the solute and solvent of the test system from the remaining data, which are finally used for predicting $\ln H_{ij}(T)$ for the test system. These predictions are then stored and compared to the respective experimental values of the withheld binary system. The training and prediction process is iteratively repeated, each time excluding a different binary system, until all systems have been omitted once. To quantify the predictive accuracy of the method, standard error scores are employed, namely the mean absolute error (MAE) and the mean squared error (MSE). A prerequisite for the leave-one-out analysis is that each component, i.e., each solute and solvent, appears in at least two binary systems within the experimental data set. This ensures that, even when one system is excluded, the method has access to some data on each component.

Besides the three parameter correlation of the temperature dependence of the Henry's law constant given in Eq. (25), also the case in which C_{ij} is set to zero was studied. The results for the MCMs based on this two-parameter correlation of the van't Hoff type are reported in Appendix C. These results are only slightly worse compared to those obtained with the three-parameter correlation, highlighting that for many systems, even a simple equation can effectively capture their temperature dependence. Therefore, more complex equations with additional parameters were not considered.

4.2.3.1 Data-Driven MCM

The purely data-driven MCM developed in the present study, hereafter referred to as *MCM-data*, relies only on the information contained in the sparse experimental data on $\ln H_{ij}(T)$ from the DDB.

For training MCM-data, a rather broad, thus uninformative, prior is employed. Specifically, the entries of $\boldsymbol{u}_i$, $\boldsymbol{v}_j$, and the biases b_i^u, b_j^v are modeled with normal distributions

centered at zero:

$$p(u_{i,k}) = \mathcal{N}(0,\, \sigma_{\mathrm{P}}),\ \text{for } k = 1...K \tag{27}$$

$$p(v_{j,k}) = \mathcal{N}(0,\, \sigma_{\mathrm{P}}),\ \text{for } k = 1...K \tag{28}$$

$$p(b_i^u) = \mathcal{N}(0,\, \sigma_{\mathrm{P,CB}}) \tag{29}$$

$$p(b_j^v) = \mathcal{N}(0,\, \sigma_{\mathrm{P,CB}}) \tag{30}$$

The standard deviations in Eqs. (27) to (30) are hyperparameters, which were set to $\sigma_{\mathrm{P}} = 1$ for $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$, and $\sigma_{\mathrm{P,CB}} = 10$ for the biases, based on findings from Chapter 4.1, in which the effects of the choice of hyperparameters were investigated systematically. It was observed that the MCMs show robust behavior to variations in the standard deviations of the priors, allowing for flexible hyperparameter selection.

The likelihood for $\ln H_{ij}^{\mathrm{exp}}$ given the features covering the entries of $\boldsymbol{u}_i$, $\boldsymbol{v}_j$, and the biases b_i^u, b_j^v for all parameters P_{ij} is modeled with a Cauchy distribution, chosen for its robustness to outliers, centered around the prediction formula of Eq. (25):

$$p(\ln(H_{ij}^{\mathrm{exp}} \,/\, \mathrm{kPa}) \,|\, \text{features}) = \mathrm{Cauchy}(A_{ij} + \frac{B_{ij}}{T \,/\, \mathrm{K}} + C_{ij} \left(\ln\left(\frac{T}{T_0}\right) + \frac{T_0 - T}{T} \right),\, \lambda) \tag{31}$$

Here, A_{ij}, B_{ij}, and C_{ij} are functions of their respective features as defined in Eq. (26), with $\lambda = 0.15$ chosen as the scale parameter for the Cauchy distribution based on preliminary studies.

4.2.3.2 Hybrid MCM

The hybrid model, referred to as *MCM-hybrid*, enhances the purely data-driven method by integrating predictions from the physics-based PSRK EoS using its latest public parameterization [76]. This hybridization strategy was introduced for the prediction of H_{ij} at 298 K in Chapter 4.1 and is therefore only briefly discussed here. The training of MCM-hybrid involves two steps: in the *pretraining step*, the physical knowledge in the PSRK data is extracted and serves as prior knowledge for the training of MCM-hybrid on experimental data in the second step, which is referred to as the *refinement step*. This process is illustrated in Fig. 15. The corresponding, very simple schematic for MCM-data is shown in Fig. C.3 in Appendix C.

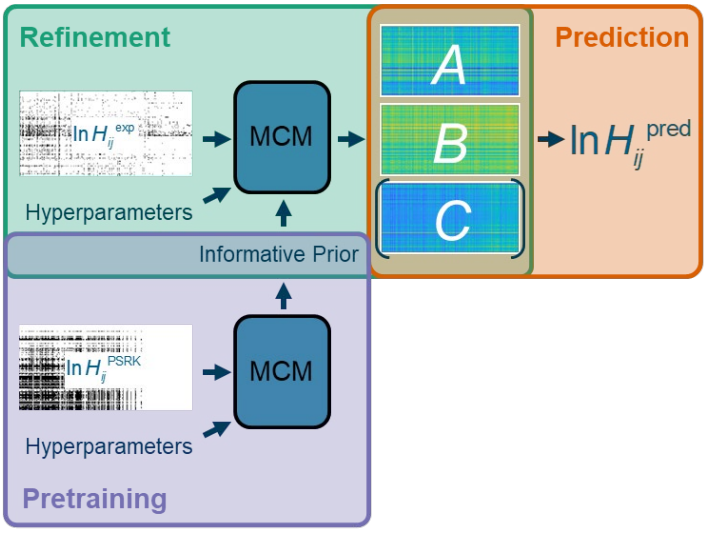


Figure 15: Schematic illustration of the prediction of $\ln H_{ij}$ with MCM-hybrid. In the pretraining step, the hyperparameters are specified, and the MCM is trained on simulated data for $\ln H_{ij}$ from PSRK. The inferred (preliminary) features are used to generate an informative prior for the refinement step, in which the MCM is trained on experimental data for $\ln H_{ij}$. The resulting (final) features are subsequently used in Eq. (26) to obtain predictions (pred) for the parameters A_{ij}, B_{ij}, and C_{ij} of the function that is used for describing the temperature dependence of $\ln H_{ij}$ (Eqs. (25) or (C.1)), from which the Henry's law constant can be calculated for any temperature.

In the pretraining step, PSRK predictions serve as training data to infer preliminary features. Due to incomplete parameter tables, PSRK predictions could only be obtained for 11,407 binary systems of the considered solutes and solvents, accounting for 23.4% of all possible combinations. Extreme outliers in the PSRK data set, i.e., predictions outside the range $-10 < \ln H_{ij}\,/\,\mathrm{kPa} < 25$, were generally excluded. These boundaries are chosen referring to the minimum and maximum values of the Henry's law constants in the experimental database. The impact of extreme outliers on applying PSRK as a reference method is discussed in more detail in the subsequent section.

PSRK enables predictions in a wide range of temperatures. However, using predictions at three distinct temperatures within the 173.15 K to 573.15 K range was found to be sufficient. The selection of temperatures for which PSRK predictions were used was system-specific, since the minimum and maximum temperature at which PSRK can be used depends on the solute and solvent. Specifically, PSRK predictions were calculated in 5 K steps within the specified temperature range for all systems, and results at low and high temperatures, for which PSRK provided no results or extreme outliers, were excluded as previously discussed. From the remaining temperature range, the predicted $\ln H_{ij}$ values at the minimum, mean, and maximum temperature were selected.

Similar to MCM-data, an uninformative normal distribution is employed for the pretraining step's prior, cf. Eqs. (27) - (30), and the likelihood in the pretraining step is modeled by a Cauchy distribution centered around the PSRK prediction, compare Eq. (31). With both the prior and the likelihood defined, the posterior, which incorporates the physical information extracted from the PSRK predictions, can be calculated through variational inference [19].

Subsequently, a normal distribution is fitted to samples from the posterior distribution of each feature, representing preliminary features that form the basis of an informative prior for the refinement step. The approach to generating an informative prior is analogous to the approach proposed in Chapter 4.1, except that the features learned here model the three parameter matrices (A_{ij}, B_{ij}, C_{ij}), which are treated independently. For transferring the informative prior to the refinement step, the means of the preliminary features are kept, while the standard deviations are scaled in such a way that their average is $\sigma_{\mathrm{P}} = 0.5$ and $\sigma_{\mathrm{P,CB}} = 5$, respectively. In general, features supported by extensive data retain smaller standard deviations, indicating higher confidence. Conversely, features with less supporting data exhibit larger standard deviations, denoting lower confidence. The distributions are then multiplied by the respective uninformative prior distribution. These adjustments take into account the information that was gained in the pretraining step in a soft form, while still ensuring that all informative prior distributions have smaller standard deviations than their uninformative counterpart.

For each of the three parameter matrices (A_{ij}, B_{ij}, C_{ij}), the resulting informative prior for the refinement step can be expressed as a normal distribution centered around the initial guess of each feature ($u^*_{i,k}$, $v^*_{j,k}$, b^{u*}_i, b^{v*}_j) with the standard deviations (σ^*_{P}, $\sigma^*_{\mathrm{P,CB}}$). These normal prior distributions are unique for each feature and are pre-trained for those solutes and solvents with at least one PSRK-predictable data point. However, MCM-hybrid is not limited to components that can be modeled with PSRK; it can also be applied to all other considered components as well. In these cases, the same uninformative prior is used as in the training of MCM-data.

In the refinement step, the features defined by the informative priors are refined through training on experimental data from the DDB [86]. Thereby, the likelihood is employed analogously to the data-driven approach, as specified in Eq. (31), with $\lambda = 0.15$ maintained throughout both the pretraining and refinement steps.

The training on experimental data in the refinement step results in the final set of features, based on the PSRK predictions *and* the experimental data. Subsequently, these final features enable predictions of $\ln H_{ij}(T)$ for any solute-solvent combination from the matrix and for any temperature by applying Eqs. (25) and (26).

4.2.4 Results and Discussion

Fig. 16a presents results for the MAE and MSE obtained in the leave-one-out analysis of both MCMs developed in the present chapter using the full data set on Henry's law constants. The scores were calculated by averaging individual system scores, ensuring an equal contribution of each binary system. In Fig. 16b, the results of the MCMs are compared with those of PSRK. The experimental data set used for this comparison is a subset of that used in Fig. 16a, as PSRK can only be used for a part (53.7%) of the systems for which data are available. Furthermore, 190 data points for 21 systems, for which PSRK yielded extreme outliers, were removed as their inclusion would have drastically deteriorated the MAE and MSE of PSRK, as highlighted in Fig. C.2 in Appendix C.

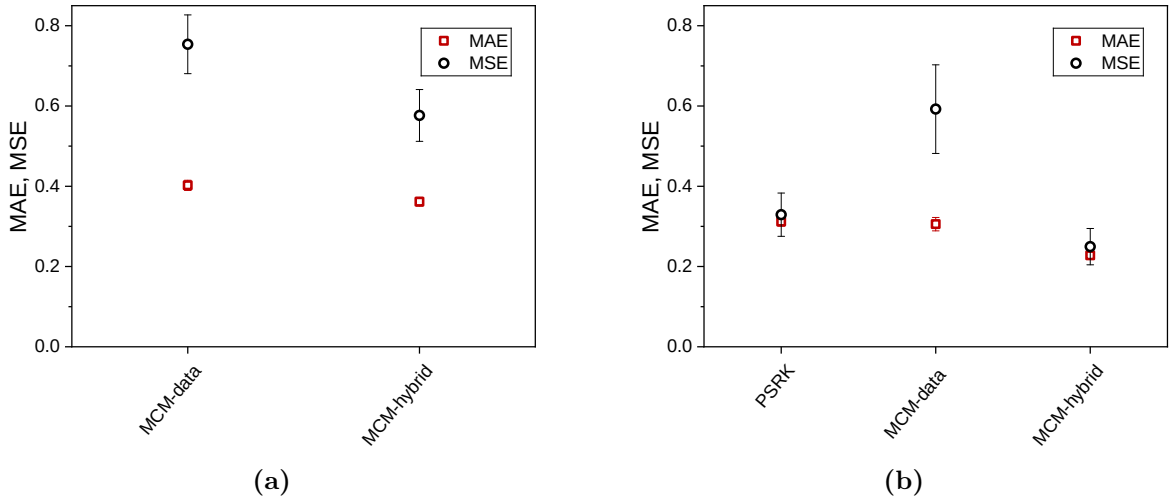


Figure 16: Mean absolute error (MAE) and mean squared error (MSE) for the prediction of $\ln H_{ij}$ averaged over all binary systems. (a) Comparison of the developed MCMs considering all data from the DDB (3,297 systems). (b) Comparison of the developed MCMs with PSRK considering only systems that can be described by PSRK (1,574 systems). 190 data points for 21 systems for which PSRK yielded extreme outliers were removed.

Fig. 16a shows that MCM-hybrid gives better results than MCM-data, as expected. The improvement in MAE is only small, but it is considerable in MSE, indicating that, as a result of the pretraining, there are fewer large errors for MCM-hybrid.

Moving from the complete database (Fig. 16a) to the subset of the data that can be described with PSRK (Fig. 16b) leads to better scores for both MCM-data and MCM-hybrid. The better results can be attributed to the fact that the subset of the data studied in Fig. 16b contains more systems with common components that have also been studied in many other systems, which is obviously an advantage for the MCM.

Going from Fig. 16a to Fig. 16b, the improvement is more pronounced for MCM-hybrid than for MCM-data. This is not astonishing as MCM-hybrid uses PSRK data in the pretraining and, therefore, implicitly carries information on systems that can be modeled with PSRK.

The comparison with PSRK in Fig. 16b shows a clear improvement in both scores for MCM-hybrid. MCM-data and PSRK have similar MAE, but MCM-data has a much higher MSE indicating that there are outliers in the predictions.

It is important to realize that the comparison of the results from PSRK to those from both MCMs carried out here is biased in favor of PSRK. The results for the MCMs were obtained from a leave-one-out analysis and are therefore strict predictions: the predicted data were not used for training the model, which is why a similar performance may be expected for new data. However, this is not the case for PSRK: it is likely that many of the experimental data that are used here for comparison were used in the training of PSRK. In comparing the MCMs with PSRK, it is also important to consider the different ranges of applicability. The MCMs can be applied for all systems of the matrix, while this is only the case for 23.4% of these systems for PSRK due to missing interaction parameters. On the other hand, PSRK is a group-contribution approach and can therefore also be applied to systems outside of the matrix, as long as their components can be built from parameterized groups. Furthermore, PSRK can, of course, do more than just predicting the Henry's law constant.

All predictions of MCM-hybrid used to calculate the scores in Fig. 16 are reported in an accompanying Excel sheet in Ref. [87]. The MCMs also provide information on the uncertainties of the predictions since a probability distribution is obtained for each $\ln H_{ij}$; these uncertainties are also reported in terms of standard deviations. To facilitate practical application of the hybrid MCM results, an additional Excel sheet contains the set of *final* parameters, i.e., the filled matrices A_{ij}, B_{ij}, and C_{ij}, for all solute-solvent combinations. Unlike the leave-one-out approach discussed earlier, these parameters are derived from training MCM-hybrid on *all* experimental $\ln H_{ij}(T)$ data. Consequently, users can directly apply these final parameters using Eq. (25) to predict $\ln H_{ij}(T)$ for all combinations of solutes and solvents in the matrix. The final parameters can also be used to calculate the enthalpy of absorption of the solute in the solvent, as discussed in Appendix C.

The total MAEs depicted in Fig. 16b can be further broken down for individual systems in the form of a histogram, which is shown in Fig. 17.

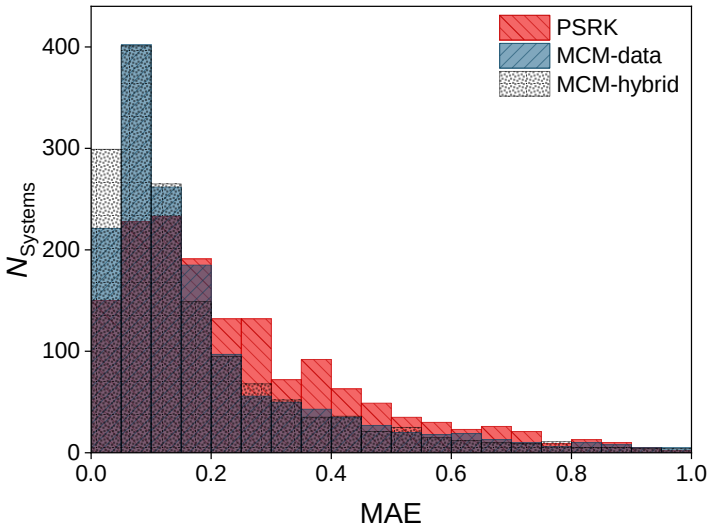


Figure 17: Histogram representation of the MAE for the predictions of $\ln H_{ij}(T)$ with PSRK, MCM-data, and MCM-hybrid considering only systems predictable by PSRK (1,574 systems). N_{Systems} is the number of binary systems predicted with a specific MAE. The shown interval in the histogram contains 96.32% (PSRK), 94.79% (MCM-data), and 96.63% (MCM-hybrid) of all considered binary systems.

Fig. 17 reveals that both MCMs achieve high accuracies (small deviations of the predictions from the experimental data) for more systems than PSRK, with MCM-hybrid outperforming MCM-data. While MCM-data predicts a larger number of systems with high precision compared to PSRK, it also exhibits more outliers falling outside the histogram's depicted range. These observations align with the findings for Fig. 16, underlining MCM-hybrid's ability to merge the strengths of both methods, leading in particular to an improved prediction of those systems that tend to be poorly predicted with PSRK or MCM-data.

Fig. 18 presents six examples of predicted temperature-dependent Henry's law constants in binary systems and compares them to experimental test data. Three of the examples can be predicted with all methods, the other systems can only be modeled with MCM-data and MCM-hybrid.

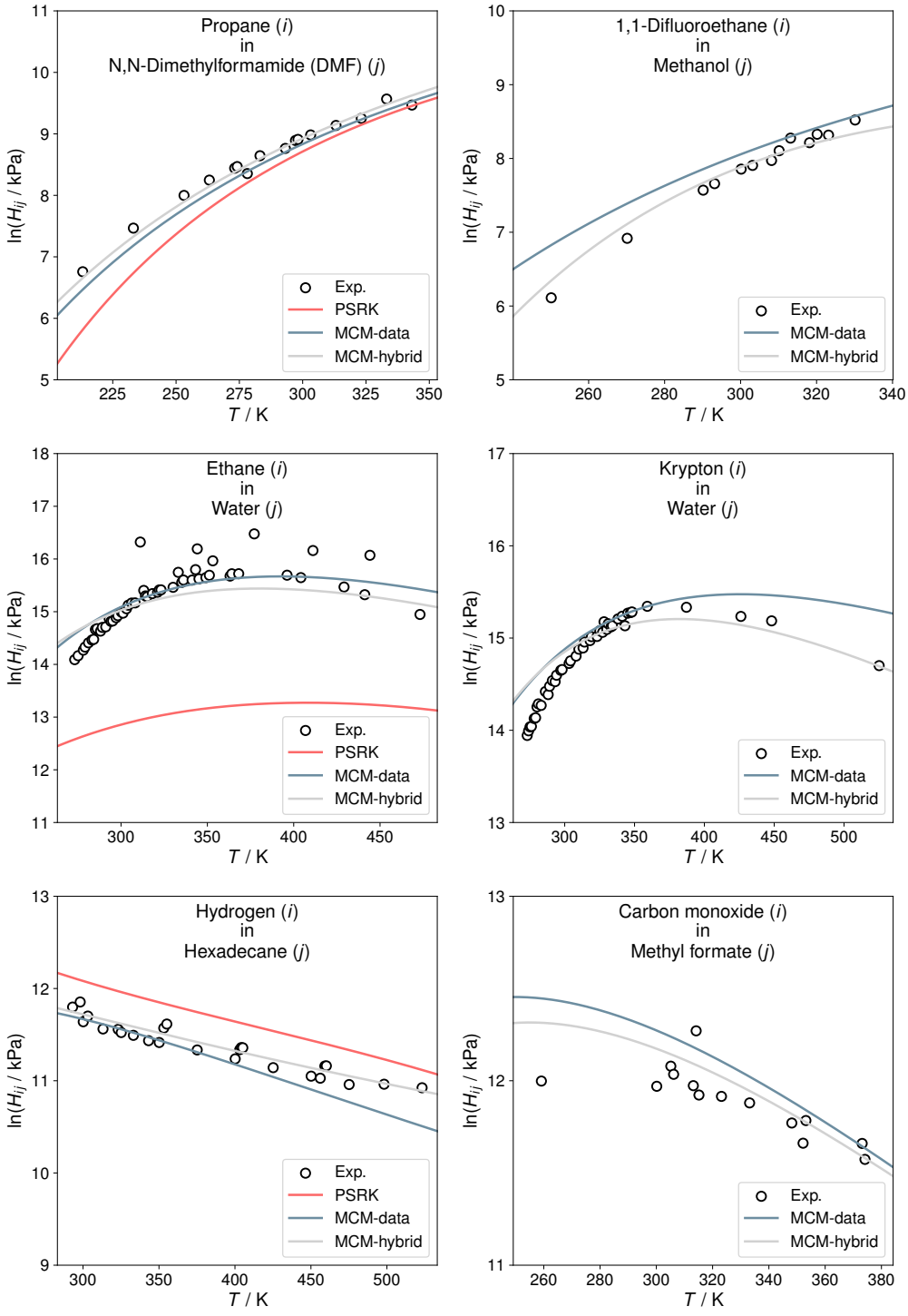


Figure 18: Temperature-dependent Henry's law constants for six binary systems predicted with PSRK, MCM-data, and MCM-hybrid (lines) and comparison with experimental (exp.) test data (circles). The results of the MCMs are true predictions obtained with a leave-one-out analysis. PSRK is only applicable for three of the systems.

The results demonstrate the ability of especially MCM-hybrid to accurately predict Henry's law constants over a wide temperature range. This study is extended in the following, where the performance of all methods for predicting $\ln H_{ij}(T)$ in different

temperature ranges is analyzed in more detail. To facilitate this analysis, the experimental data set is divided into eight temperature bins, with the MAE calculated for each bin as shown in Fig. 19.

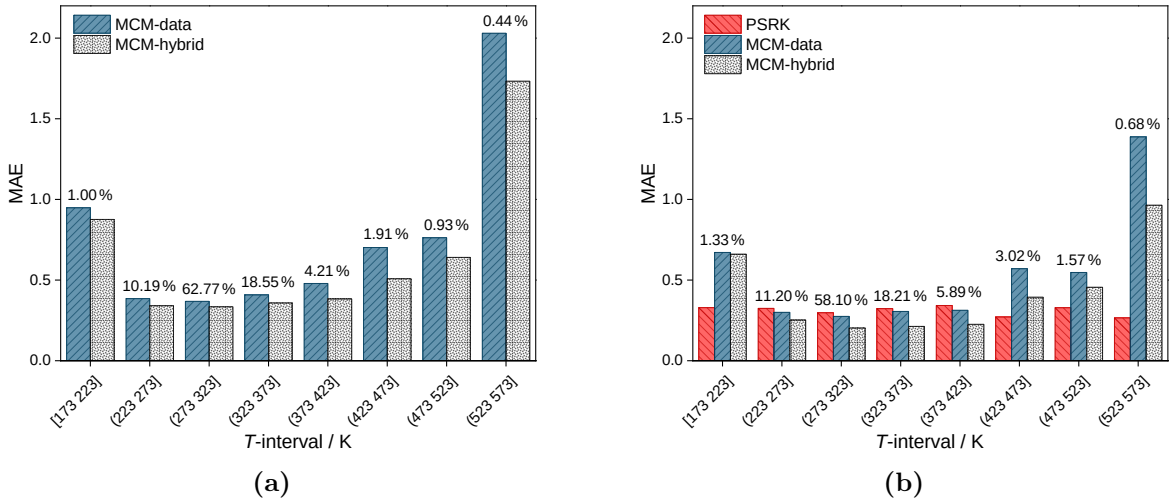


Figure 19: MAE of the predictions for $\ln H_{ij}(T)$ with PSRK, MCM-data, and MCM-hybrid as a function of the temperature averaged over all binary systems in each temperature bin. Percentages denote the fraction of experimental data points in the respective temperature range. (a) Considering all data from the DDB [86]. (b) Considering only systems that can be described by PSRK.

Fig. 19 reveals PSRK's consistent performance across all temperature ranges, exhibiting relatively uniform MAEs. In contrast, the MCMs' predictive performance depends on the temperature, with MCM-data showing a similar trend but inferior predictive accuracy compared to MCM-hybrid. This variation in the performance of the MCMs aligns with the distribution of the training data, which is disproportionately concentrated in the ambient temperature range, as highlighted in Fig. 19 and Fig. C.1 in Appendix C. This underscores the critical importance of obtaining reliable experimental training data at different temperatures for further improving the MCMs. PSRK is more reliable in temperature ranges for which there are little experimental data, i.e., at extreme temperatures. However, despite the uneven data distribution, MCM-hybrid significantly outperforms PSRK in a wide temperature range (223 – 423 K), in which 93.4% of the experimental data lie.

4.2.5 Conclusions

In this chapter, two matrix completion methods (MCMs) based on Bayesian machine learning are introduced to predict temperature-dependent Henry's law constants $H_{ij}(T)$

in binary mixtures of solutes i in pure solvents j. As the data depend on three variables (i, j, and T), a direct application of matrix completion approaches is impossible. Therefore, physical relations are employed to model the temperature dependence of H_{ij}, whose system-specific but temperature-independent parameters can be organized in a matrix, facilitating matrix completion.

Furthermore, two variants are introduced: MCM-data and MCM-hybrid. While MCM-data is purely data-driven and trained solely on the scarce available experimental data on $H_{ij}(T)$ from the Dortmund Data Bank [86], MCM-hybrid additionally incorporates results from the physics-based Predictive Soave-Redlich-Kwong (PSRK) equation-of-state [76] in a pretraining step. The predictions from the MCMs are evaluated using leave-one-out analysis. As expected, the pretrained MCM-hybrid yields better predictions. MCM-hybrid also outperforms PSRK. Both MCMs show an excellent accuracy over a wide temperature range, which only declines for extreme temperatures (below 223 K and above 423 K), for which only few data are available for training. Using the MCMs, H_{ij} and the enthalpy of absorption h_{ij}^{abs} can now be predicted for all combinations of the considered 122 solutes and 399 solvents over a wide temperature range. The new MCMs can be updated easily when new data become available. This chapter also underlines that the idea of matrix completion can be applied successfully for predicting basically any binary physicochemical property, either in a purely data-driven way, or in a hybrid way by including physicochemical knowledge, which further enhances the performance of the resulting models.

4.3 Activity Coefficients at Infinite Dilution

4.3.1 Introduction

Reliable prediction methods for thermodynamic properties of mixtures are essential for the design and optimization of many processes in chemistry and chemical engineering. A particularly important property is the activity coefficient, which describes the deviation from the ideal mixture and is widely used for modeling reaction and phase equilibria in mixtures. Common physical prediction methods for activity coefficients include group-contribution (GC) models, such as UNIFAC [12, 41] and modified UNIFAC (Dortmund) (mod. UNIFAC) [13, 42], as well as models based on quantum chemistry like COSMO-RS [34, 43, 44] and COSMO-SAC-dsp [47].

While these physical methods are well established, numerous alternative methods for predicting activity coefficients have recently been developed based on machine learning (ML). Some of them learn exclusively from available experimental data [1, 8, 48, 89], others are hybrid methods that combine the strengths of ML with those of physics [2, 9, 10, 49–54, 57, 82–85, 90]. An example for such a hybrid model is the so-called Whisky method [9], which belongs to the class of matrix completion methods (MCMs). It was developed for the prediction of activity coefficients of solutes i at infinite dilution in solvents j γ_{ij}^{∞} at 298 K in unstudied binary mixtures. The MCM exploits the fact that the properties of binary mixtures can be stored conveniently in matrices, which are only sparsely occupied by experimental data in all relevant cases. In the Whisky method, a Bayesian approach is used to complete the matrix, exploiting similarities between components that are learned in the training. This training involves two steps, named in analogy to Whisky production: a distillation step, in which information from mod. UNIFAC predictions is distilled into prior knowledge, followed by a maturation step, in which the model is refined using experimental data. This hybrid approach, which combines the physics-based mod. UNIFAC with a data-driven MCM, has demonstrated superior performance compared to a purely data-driven MCM and the mod. UNIFAC model alone [9]. For more technical details on the Whisky method, see Ref. [9].

As an alternative to the Whisky method, the so-called similarity-based method (SBM) (cf. Chapter 3) for predicting activity coefficients at infinite dilution has been developed. The SBM is based on the idea that similar mixtures should have similar properties, following the ancient alchemistic knowledge "similia similibus solvuntur". Consequently, the SBM uses the available experimental data for activity coefficients in similar mixtures to predict the activity coefficients in an unstudied mixture of interest by simply averaging the data for similar mixtures. At its core, the SBM calculates similarities between mixture components to determine mixtures for which data are available and which are

sufficiently similar to the (unstudied) mixture of interest so that the data can be used for the prediction.

It is clear that the accuracy and the range of applicability of the SBM are inversely correlated: the higher the demanded accuracy, the stricter the required similarity, and the lower the chance of finding sufficiently similar mixtures in a given data set. In the SBM, the same chemical descriptors as they are used in the COSMO models [34, 55] are used for defining a similarity score. The SBM achieves a high prediction accuracy, often within typical experimental uncertainties, whenever sufficient similar data are available.

In this chapter, a novel model combining the Whisky approach with the SBM is proposed. Specifically, synthetic data from the SBM are integrated in the training process of the Whisky method, basically doubling the amount of training data in Whisky's maturation step. Therefore, a strict similarity criterion has been applied in the SBM, leading to precise predictions (at the cost of the number of mixtures for which the SBM yields predictions). The novel model, which is called Blended Whisky, thereby combines the strengths of both underlying methods: the Whisky method enables a broad scope by filling the entire matrix of missing data, while the SBM contributes precise predictions, which act as a powerful substitute for actual experimental data, increasing the prediction accuracy. This synergy leads to a more robust and accurate predictive framework that consistently outperforms its predecessors.

In addition to introducing the Blended Whisky method, this chapter discusses and emphasizes the implications of model assessment and training data design for ML models, particularly MCMs. By analyzing the correlations between the training data quantity and type (synthetic or experimental), the similarity between the mixtures of interest and those in the training set, and the overall model performance, fundamental insights are obtained for the efficient training of MCMs, laying the foundation for their advancement, particularly in data-sparse regions.

4.3.2 Development of the Blended Whisky Method

The Blended Whisky method is an MCM that combines two approaches for predicting isothermal activity coefficients at infinite dilution γ_{ij}^{∞} in binary mixtures: the Whisky method [9], a probabilistic MCM using experimental data and synthetic data from the physical mod. UNIFAC [13] model for training, and the SBM (cf. Chapter 3), which makes predictions using similarity-based imputation. Details on the Whisky method are provided in Ref. [9].

The Blended Whisky method developed here is a probabilistic model that calculates logarithmic activity coefficients at infinite dilution as follows:

$$\ln \gamma_{ij}^{\infty} = \boldsymbol{u}_i \cdot \boldsymbol{v}_j + \varepsilon_{ij} \tag{32}$$

where $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$ are feature vectors of the solute i and solvent j, respectively, and ε_{ij} is a random variable that captures experimental noise and model inaccuracies. The solute- and solvent-specific feature vectors can be aggregated into two feature matrices, $\boldsymbol{U}$ and $\boldsymbol{V}$, representing the learned characteristics of all solutes and all solvents, respectively, in the data set.

Blended Whisky is a Bayesian method (cf. Chapter 2) and, in contrast to simple MCMs [8], which infer the component features ($\boldsymbol{u}_i$ and $\boldsymbol{v}_j$) only from the available experimental data in a single step, it (analogously to the Whisky method [9]) is based on a two-step approach with different data sources. Fig. 20 shows a schematic of the training process of Blended Whisky.

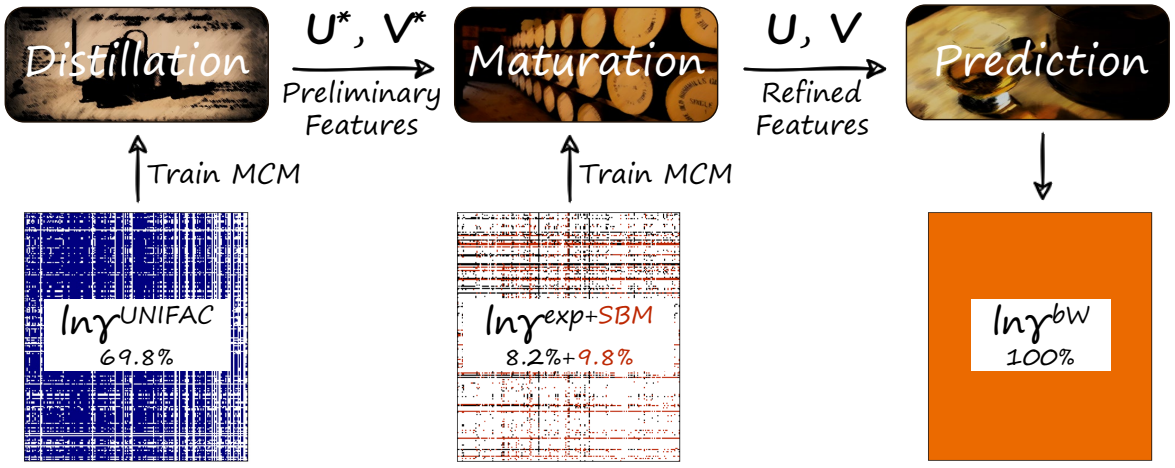


Figure 20: Schematic illustration of the Blended Whisky method. In the distillation step, an MCM is trained on mod. UNIFAC predictions for $\ln \gamma_{ij}^{\infty}$ at 298 K. The thereby fitted MCM parameters, stored in component feature matrices $\boldsymbol{U}^*$ and $\boldsymbol{V}^*$, are used as informative prior for the maturation step, where a second MCM is trained on the available experimental data augmented with SBM predictions. The final MCM parameters are used for making predictions for unstudied $\ln \gamma_{ij}^{\infty}$.

In the first step, the distillation step, knowledge from mod. UNIFAC captured in its predictions is distilled and this knowledge is stored in a first set of component features. Thereby, a rather broad, uninformative prior is used for each feature, specifically, a normal distribution with a mean of $\mu = 0$ and a standard deviation of $\sigma = 0.8$. As likelihood, a Cauchy distribution with scale parameter $\lambda = 0.15$ centered around the product of preliminary feature vectors is used:

$$p(\ln \gamma_{ij}^{\infty,\text{mod. UNIFAC}} \,|\, \boldsymbol{u}_i^*, \boldsymbol{v}_j^*) = \text{Cauchy}(\boldsymbol{u}_i^* \cdot \boldsymbol{v}_j^*,\, \lambda) \tag{33}$$

For all components for which mod. UNIFAC predictions were available during the distillation step, the resulting posterior mean is retained and is, in combination with a standard deviation of $\sigma = 0.5$, used as informative prior for the subsequent maturation step. For all other components, i.e., those for which mod. UNIFAC can not predict the activity coefficients in the distillation step, a broader normal distribution with a mean of $\mu = 0$ and a standard deviation of $\sigma = 3$ is used as the prior for the maturation step.

In the maturation step, the features obtained from the distillation step are refined by training on experimental data ($\ln \gamma_{ij}^{\infty,\text{exp}}$) and, in contrast to the Whisky method, synthetic training data obtained from the SBM, as described below. This way, the otherwise sparse experimental training set is substantially augmented with synthetic data of high quality (cf. Chapter 3). The likelihood in the maturation step follows a Cauchy distribution with a scale parameter of $\lambda = 0.15$ for the experimental data and $\lambda = 0.2$ for the SBM data.

The synthetic data used for augmenting the training data in the maturation step of the Blended Whisky method were obtained using the SBM approach from Chapter 3, which is based on a similarity score S derived from quantum-chemically calculated σ-profiles. The SBM makes predictions for $\ln \gamma_{ij}^{\infty}$ by averaging the experimental data from mixtures that are similar to the one of interest. Thereby, a *similar mixture* is defined as one with the same solute i (solvent j) and a different solvent n (solute m) that has a similarity score with the solvent of interest j (solute of interest i) higher than a threshold, which was set to 0.93, i.e., $S_{nj} > 0.93$ ($S_{mi} > 0.93$). The similarity score has values between 0 (no similarity) and 1 (full similarity). The choice of the threshold value $\xi = 0.93$ indicates that a high degree of similarity is required, leading to reliable predictions of the γ_{ij}^{∞} with the SBM. The downside is that choosing a high value of ξ leads to the fact that there will be only a few mixtures for which sufficiently similar mixtures for which data exist can be found. In this case, the SBM with $\xi = 0.93$ yielded only additional results for 9.3% of the entries of the matrix. However, since only for 8.6% of the entries experimental data were available, the database for the maturation step of the Blended Whisky method could be more than doubled. Whenever an experimental value and an SBM prediction were available, the experimental value was used.

The Blended Whisky method was trained on 30,597 synthetic data points from mod. UNIFAC in the distillation step, and 3,568 experimental data points along with 4,277 synthetic data points from the SBM in the maturation step. All experimental $\ln \gamma_{ij}^{\infty,\text{exp}}$ data were taken from the Dortmund Data Bank (DDB) [38]; in total, 221 different solutes

and 198 different solvents were considered. These experimental data and the synthetic training data obtained from the SBM are identical to the ones in Chapter 3.

Blended Whisky was implemented in the probabilistic programming language Stan [20] and trained using Gaussian mean-field variational inference. The hyperparameters, namely the standard deviations of the prior and the scale parameters of the likelihood, were determined through preliminary studies.

The performances of the studied MCMs (MCM-data, Whisky, and Blended Whisky) and the SBM were evaluated using a leave-one-out analysis. Thereby, for each binary mixture with available experimental data ($\ln \gamma_{ij}^{\infty,\mathrm{exp}}$), the corresponding data point was excluded from the training set. The remaining data were then used to generate predictions for the excluded mixture. This approach ensures that the predictions are independent of the experimental data for that particular mixture, providing a more rigorous test of the predictive performance of the method. All calculations were performed using Matlab [23].

4.3.3 Results and Discussion

4.3.3.1 Overall Performance of Blended Whisky

In Fig. 21, the performance of the Blended Whisky method for predicting $\ln \gamma_{ij}^{\infty}$ at 298 K obtained by a leave-one-out analysis is shown in terms of the mean absolute error (MAE) and the mean squared error (MSE). It is compared to the performances of the building blocks of Blended Whisky, the SBM (cf. Chapter 3), and the Whisky method [9], as well as that of a purely data-driven MCM (MCM-data) [8] and the physical benchmarks mod. UNIFAC [13], COSMO-SAC [46], and COSMO-SAC-dsp [47]. The results are plotted as a function of the number N of predictable data points in the experimental data set, containing N_{max} = 3568 data points. For COSMO-SAC and COSMO-SAC-dsp, the implementations by Bell et al. [55] were used.

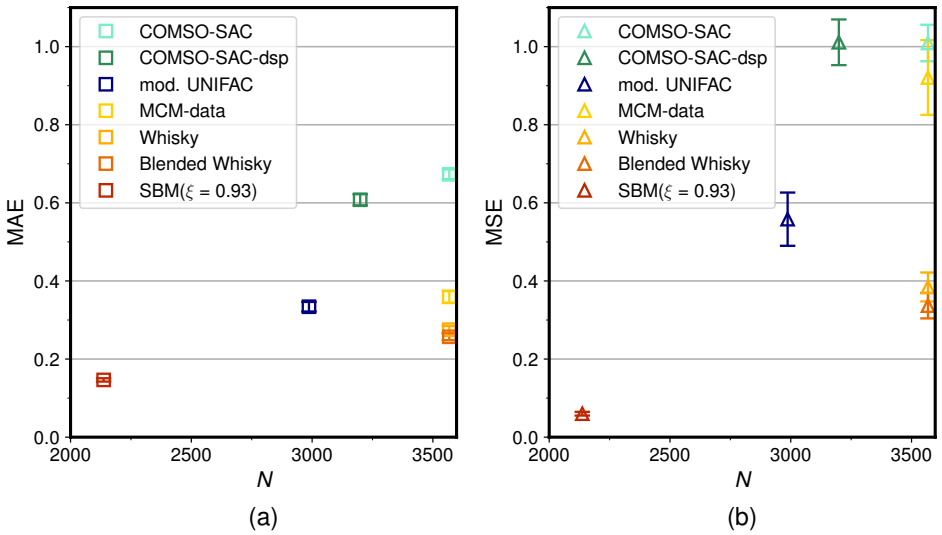


Figure 21: Mean absolute error (MAE, panel a) and mean squared error (MSE, panel b) of the predictions of $\ln \gamma_{ij}^{\infty}$ for the Blended Whisky method as a function of the number N of predictable data points in the data set. For comparison, the results of mod. UNIFAC, COSMO-SAC, COSMO-SAC-dsp, MCM-data, the Whisky method, and the SBM(ξ = 0.93) are shown. Error bars denote standard errors of the means.

The SBM(ξ = 0.93) achieves the highest prediction accuracy in both error scores; however, its scope is the smallest of all methods compared, as it is limited by the requirement for training data on mixtures similar to those of interest. Mod. UNIFAC offers broader applicability but at the cost of reduced accuracy. COSMO-SAC-dsp extends the scope further, albeit with even lower prediction accuracy. COSMO-SAC, MCM-data, Whisky, and Blended Whisky can predict all test data points. Among them, COSMO-SAC exhibits the poorest accuracy, while the Blended Whisky method achieves the highest accuracy (MAE = 0.26, MSE = 0.34), slightly outperforming the Whisky method (MAE = 0.28, MSE = 0.38). Although this overall reduction of the error scores may seem small, it was achieved without additional data compared to its predecessors, as both the SBM and Whisky are trained on the same database.

Notably, both hybrid MCMs (Whisky and Blended Whisky) significantly reduce outliers in their predictions, as evidenced by their lower MSE values than MCM-data. In contrast, mod. UNIFAC exhibits some extreme outliers, which have even been excluded from the error score calculations in Fig. 21. Appendix A provides a detailed analysis of these outliers. Remarkably, despite including these outliers during the training of both the Whisky and Blended Whisky methods in their distillation steps, the MCMs still achieve high prediction accuracies, outperforming mod. UNIFAC, demonstrating the robustness of the hybrid models. Of the benchmark methods, only the SBM(ξ = 0.93)

achieves a lower MAE than the Blended Whisky method; however, this score is based on only the 60% of test data that the SBM($\xi = 0.93$) can predict.

4.3.3.2 Influence of Training Data on Predictive Performance

In the following, two factors that influence the predictive performance of the MCMs are systematically examined. The first factor is the amount of experimental training data for each solute i and solvent j, corresponding to the number of data points in each row and column of the experimental data matrix. A binary mixture $i + j$ is thus characterized by the number of data points available for solute i and the number of data points available for solvent j. Only the smaller of these two values is used, representing the less frequently measured component, and it is denoted as N_{min} in the following.

In Fig. 22, it is analyzed how the prediction accuracy correlates with N_{min}. It shows the MAE and MSE of all studied methods (excluding the SBM($\xi = 0.93$) due to its limited scope) on a shared test data set containing only mixtures predictable by all methods. Mixtures with rarely measured components ($N_{\mathrm{min}} \leq N_{\mathrm{cutoff}}$) and frequently measured components ($N_{\mathrm{min}} > N_{\mathrm{cutoff}}$) are differentiated, using a cutoff value of $N_{\mathrm{cutoff}} = 5$.

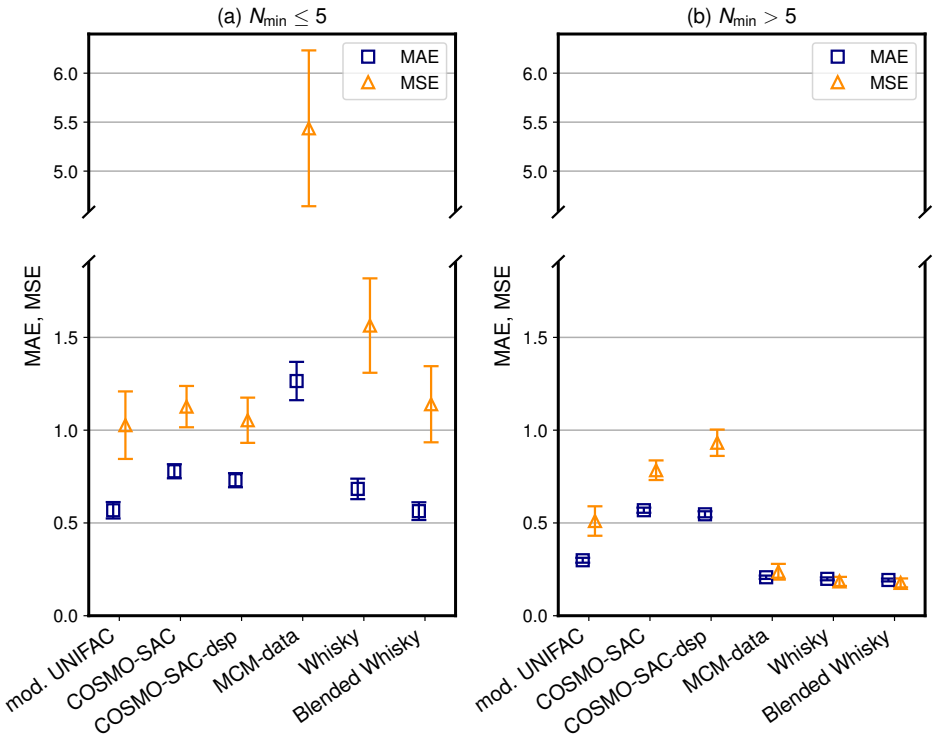


Figure 22: Mean absolute error (MAE) and mean squared error (MSE) of predictions of $\ln \gamma_{ij}^{\infty}$ for the Blended Whisky method in comparison to Whisky, MCM-data, mod. UNIFAC, COSMO-SAC, and COSMO-SAC-dsp. Error bars denote standard errors of the means. (a) Binary mixtures containing components with five or less studied mixtures in the data set ($N_{\text{min}} \leq 5$; 361 data points). (b) Binary mixtures where both components were studied in more than five mixtures in the data set ($N_{\text{min}} > 5$; 2,418 data points).

All methods depicted in Fig. 22 show higher average accuracy for mixtures with $N_{\text{min}} > 5$ as for mixtures with $N_{\text{min}} \leq 5$. The observed improvement in accuracy for the MCMs with increasing N_{min} is expected, given that these models are explicitly trained on the available experimental data in the database and rely on learning the component-specific features from these data; hence, more available data for a specific component can be expected to increase the quality of the learned features. In contrast, the physical benchmark methods (mod. UNIFAC, COSMO-SAC, and COSMO-SAC-dsp) have been evaluated "as-is", using their published parameters, without any additional training or fine-tuning on the data set used in this chapter. Since the training sets for these methods are not fully disclosed, it is plausible that they may have been optimized or validated using data sets that overlap with frequently measured components in the database used here, which could explain their improved performance for those mixtures. Alternatively, the mixtures in Fig. 22a may pose greater challenges for all methods due to higher molecular complexity or less predictable interaction effects.

For mixtures with only frequently studied components ($N_{\text{min}} > 5$), all three MCMs significantly outperform the three physical benchmark models. In contrast, for the mixtures with $N_{\text{min}} \leq 5$, mod. UNIFAC and the Blended Whisky method prove to be the best-performing approaches. Especially when comparing the performance of Blended Whisky to Whisky and MCM-data, the positive impact of integrating synthetic data from mod. UNIFAC *and* the SBM($\xi = 0.93$) into the training process becomes evident, particularly reducing the MSE.

While so far, only the influence of data quantity for specific solutes or solvents has been investigated, the influence of whether data for *similar* mixtures are available on the performance of the MCMs is now investigated. This distinction is not relevant to the physical benchmarks, as their performance does not depend on the presence of similar mixtures in the training data; hence, they are not further discussed in the following. The impact of similar mixtures in the training data on the performance of the different MCMs for mixtures with $N_{\text{min}} > 5$ is first explored, as shown in Fig. 23. For this purpose, the data set from Fig. 22b is split by categorizing each mixture according to the availability of at least one similar mixture (defined by a similarity score above a threshold of $\xi = 0.93$) in the training set. All mixtures meeting this condition can also be predicted using the SBM($\xi = 0.93$), which also requires at least one data point of a similar mixture for prediction. Consequently, the error scores of the SBM($\xi = 0.93$) are included in Fig. 23a.

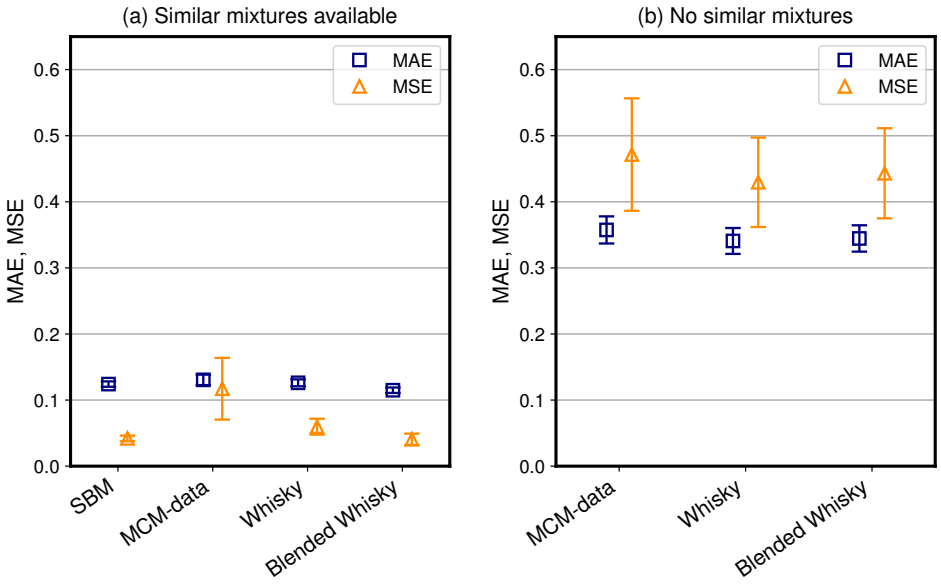


Figure 23: Mean absolute error (MAE) and mean squared error (MSE) of predictions of $\ln \gamma_{ij}^{\infty}$ for the Blended Whisky method in comparison to Whisky and MCM-data, focusing only on mixtures where both components were studied in more than five mixtures in the data set ($N_{\text{min}} > 5$; 2,418 data points). Error bars denote standard errors of the means. (a) Binary mixtures for which experimental training data of similar mixtures are available (1,600 data points); similar mixtures are defined by a similarity score above a threshold of $\xi = 0.93$. An additional comparison with the SBM($\xi = 0.93$) is performed here. (b) Binary mixtures for which no experimental data with similar mixtures are available (818 data points).

Fig. 23 shows that similar mixtures in the training set significantly enhance the performance of all MCMs, emphasizing that a large amount of data alone is not sufficient for efficient training of data-driven methods; the similarity between the unstudied mixture of interest and the studied ones in the training set is, hence, a crucial factor for prediction accuracy. The performance of the different MCMs is generally similar here, although MCM-data performs slightly worse than the hybrid MCMs (Whisky and Blended Whisky), especially in terms of MSE, for both cases in Fig. 23.

Fig. 24 shows the influence of available similar mixtures in the training set on the prediction accuracy for mixtures with rarely measured components ($N_{\text{min}} \leq 5$; cf. Fig. 22a).

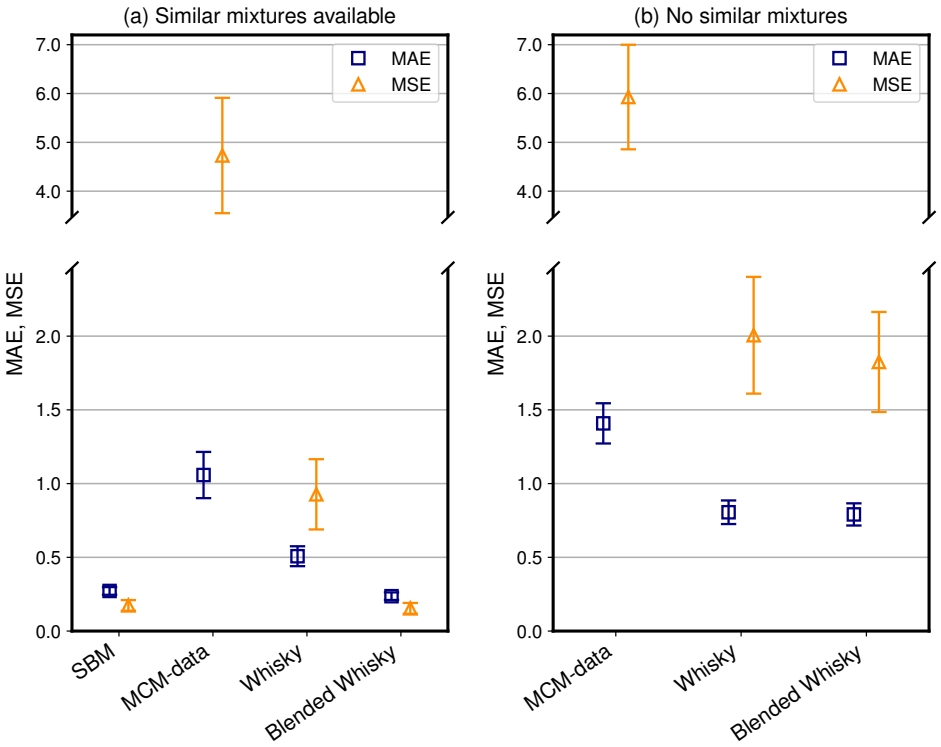


Figure 24: Mean absolute error (MAE) and mean squared error (MSE) of predictions of $\ln \gamma_{ij}^{\infty}$ for the Blended Whisky method in comparison to Whisky and MCM-data, focusing only on mixtures containing components with five or less studied mixtures in the data set ($N_{\text{min}} \leq 5$; 361 data points). Error bars denote standard errors of the means. (a) Binary mixtures for which experimental training data of similar mixtures are available (148 data points); similar mixtures are defined by a similarity score above a threshold of $\xi = 0.93$. An additional comparison with the SBM($\xi = 0.93$) is performed here. (b) Binary mixtures for which no experimental data with similar mixtures are available (213 data points).

Fig. 24 shows again that the MCMs benefit strongly from the availability of training data for similar mixtures. Again, MCM-data performs worse than the hybrid methods and Blended Whisky yields better results than Whisky. This is especially true if similar mixtures are available, because in that case, the SBM($\xi = 0.93$) yields excellent results and effectively supports Blended Whisky via the accurate synthetic data in the distillation step. The Blended Whisky method (MAE=0.24, MSE=0.15) even slightly surpasses the SBM($\xi = 0.93$) (MAE=0.27, MSE=0.17) while covering a significantly broader scope.

4.3.4 Conclusions

In this chapter, the Blended Whisky method has been developed, a hybrid matrix completion method (MCM) that successfully combines the strengths of two previously developed approaches, the Whisky method and the similarity-based method (SBM), to predict γ_{ij}^{∞} with high accuracy and broad scope. By incorporating synthetic data from the SBM as supplementary training data in the Whisky method's framework, the Blended Whisky method achieves superior performance compared to physical benchmarks and its predecessors, especially in data-sparse regions that previously challenged the Whisky method.

Furthermore, a detailed analysis has been carried out to examine how the training data affects the accuracy of different MCMs. When only limited experimental training data are available for the components that make up the mixtures of interest, the prediction accuracy of the MCMs suffers but can be significantly improved by pre-training on predictions from mod. UNIFAC. Additionally augmenting the experimental training set with synthetic data from the SBM, as used in the proposed Blended Whisky method, leads to further improvements. On the other hand, the sheer amount of training data is not everything that is important to achieve very high prediction accuracy. The training data must contain information on mixtures that are similar to the target mixtures. These insights are valuable for selecting training data for MCMs and other data-driven prediction methods and pave the way for developing targeted design-of-experiments (DOE) strategies. This chapter also shows that using similarity measures is helpful for ML studies of pure components and mixtures in different ways: in analyzing and selecting training data as well as in assessing uncertainties of the predictions.

4.4 Diffusion Coefficients at Infinite Dilution

4.4.1 Introduction

Diffusion plays a central role in many processes in nature and industry. Despite this, experimental data on diffusion coefficients are astonishingly scarce. In the present thesis, this topic is addressed for diffusion coefficients of a solute i highly diluted in a solvent j, which are particularly important both for practical and theoretical reasons, by developing novel methods for their prediction.

In general, *mutual diffusion* must be distinguished from *self-diffusion*. Mutual diffusion refers to the motion of *collectives* of molecules of different components in a mixture, and is directly relevant for describing technical processes. Self-diffusion, on the other hand, refers to the Brownian motion of *individual* molecules, and is defined for pure components as well as for mixtures.

There are two common approaches for describing mutual diffusion: the Fickian and the Maxwell-Stefan approach. Only binary mixtures are studied here, so that the following discussion is limited to this case. The Fickian diffusion coefficient D_{ij} and the Maxwell-Stefan diffusion coefficient $Đ_{ij}$ in a binary mixture $(i + j)$ are related by Eq. (34):

$$D_{ij} = Đ_{ij}\Gamma_{ij} \tag{34}$$

where Γ_{ij} is the thermodynamic factor. Both D_{ij} and $Đ_{ij}$ are in general functions of temperature, pressure, and composition. The influence of pressure on diffusion coefficients in liquids is small and neglected here, and the temperature is fixed to 298±1 K, because this temperature is of particular interest and more data on diffusion coefficients are available for 298 K than for any other temperature. Furthermore, only diffusion coefficients at infinite dilution are considered here, for which the thermodynamic factor is unity. Moreover, mutual and self-diffusion are identical at the state of infinite dilution by definition. Hence, at infinite dilution, the three cases discussed here need not be distinguished:

$$D_{ij}^{\infty} = Đ_{ij}^{\infty} = D_i^{\infty} \tag{35}$$

Here, i refers to the infinitely diluted component, j to the solvent, the index $^{\infty}$ to the state of infinite dilution, and D_i to the self-diffusion coefficient of component i. Only the symbol D_{ij}^{∞} will be used in the following.

Information on D_{ij}^{∞} is directly relevant in problems in which the diffusing component is diluted. Furthermore, there are methods to estimate $Đ_{ij}$ at finite concentrations from the respective values at infinite dilution, i.e., of $Đ_{ij}^{\infty}$ and $Đ_{ji}^{\infty}$, most notably that of Vignes [91]

for binary mixtures, which has also been extended to multi-component mixtures where experimental data on diffusion coefficients are lacking almost completely [92].

Several correlations for the prediction of D_{ij}^{∞} in binary liquid mixtures have been proposed in the literature [93], of which the most commonly used ones are those of Wilke and Chang (1955), Reddy and Doraiswamy (1967), Tyn and Calus (1975), and the Stokes-Einstein Gierer-Wirtz Estimation (SEGWE) of Evans at al. (2018) [94–97]. They are all empirical extensions of the Stokes-Einstein equation [98] and may therefore be classed as semiempirical models.

A large number of further semiempirical models for the prediction of D_{ij}^{∞} in binary liquid mixtures or extensions upon the previously mentioned ones exist in the literature, but most of them are either less general (in the scope of the components that can be modeled by them) or less accurate than these [99]. Power-law models, which have also been applied in the literature for modeling diffusion coefficients [100–102], suffer from a similar restriction in generality as they must be "calibrated" to a specific substance group, and they depend strongly on the type of components investigated. For a more detailed discussion of such approaches and their delimitation from the semiempirical models investigated here, the author refers to the review of Evans [103].

As an alternative to physical and semiempirical prediction methods for thermophysical properties in general, data-driven approaches from machine learning (ML) are presently gaining much attention [1, 104–106]. In most of the respective works, ML algorithms are thereby used for correlating thermophysical properties of pure components to a set of selected pure-component descriptors in a supervised manner. As such, most of these approaches can be classified as quantitative structure-property relationships (QSPR) [67].

Descriptor-based methods of the QSPR type can also be used for predicting mixture properties, and of course also for the prediction of diffusion coefficients. In particular, artificial neural networks (ANNs) have been used successfully in QSPR approaches by several authors [107–110], however, these studies were often restricted to specific mixtures, such as diffusion in water [109, 110] or diffusion in hydrocarbon mixtures [107]; general-purpose models for the prediction of diffusion coefficients at infinite dilution based on ML methods are still missing to date.

An interesting class of *unsupervised* ML algorithms for the prediction of thermophysical properties of mixtures in general, and of D_{ij}^{∞} in particular, are matrix completion methods (MCMs), which are already established in recommender systems, e.g., for providing suitable movie recommendations to customers of streaming providers [111, 112]. The relevance of MCMs for predicting thermophysical properties of binary mixtures has only been realized recently [1, 8]. In particular, they have been applied very successfully for

predicting activity coefficients and Henry's law constants [8–10, 82, 84]. In the present chapter, the MCM approach is extended to the prediction of diffusion coefficients.

The experimental database of liquid-phase diffusion coefficients at infinite dilution D_{ij}^{∞} in binary mixtures at 298.15±1 K consolidated by Großmann et al. [83] is used in the following. These data can be represented in an $m \times n$ matrix, in which the rows represent the solutes (m = 208) and the columns represent the solvents (n = 51). However, only 353 of the 10,608 elements of that matrix are occupied with experimental data, corresponding to 3.3%. For the rest, experimental data are missing.

In this thesis, a data-driven MCM for the prediction of D_{ij}^{∞}, which is trained only on the few available experimental data points on D_{ij}^{∞} from the database, as well as two hybrid MCMs, which combine the semiempirical SEGWE method [97] with the data-driven MCM in different ways, have been developed. The performance of the MCMs have been systematically evaluated and compared to four widely applied semiempirical methods for the prediction of D_{ij}^{∞}, namely those of Wilke and Chang [94], Reddy and Doraiswamy [95], Tyn and Calus [96], and SEGWE [97].

All MCMs presented in this chapter are collaborative-filtering approaches that learn only from the available data for the mixture property D_{ij}^{∞}, but do not require information on additional descriptors of the solutes and solvents, which is in contrast to supervised QSPR methods [27]. The predictions of the MCMs were compared to each other and to the results from the established semiempirical models.

4.4.2 Database

As a result of the consolidation procedure of Großmann et al. [83], a database on D_{ij}^{∞} containing 353 data points for 208 solutes i and 51 solvents j was obtained. The database is represented in Fig. 25 in matrix form, where the rows represent the solutes i and the columns represent solvents j, both of which are simply identified by numbers. The value for D_{ij}^{∞} is indicated by the color of the respective matrix entry. The order of the solutes and solvents does not have a meaning but was chosen to be ascending with regard to the DDB identification numbers; Table D.1 in Appendix D gives a list with the names of all considered solutes and solvents and their identification numbers. The numerical values for D_{ij}^{∞} are given in Ref. [83]. The values are censored in cases in which they have been directly adopted from the DDB and licensing restrictions prohibit their publication.

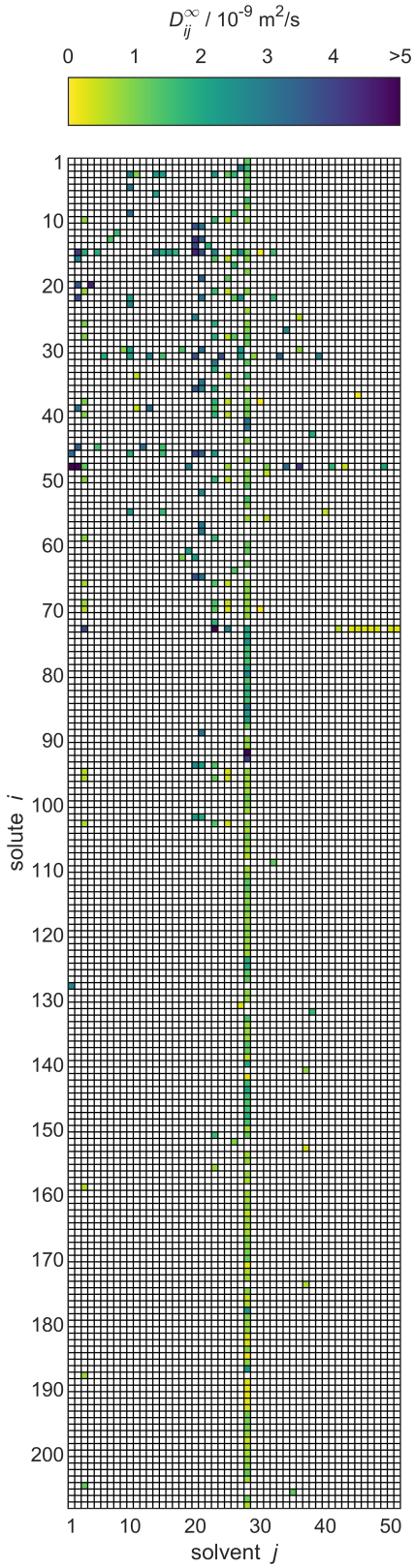


Figure 25: Overview of the experimental data for liquid-phase diffusion coefficients D_{ij}^{∞} of solutes i in solvents j at infinite dilution at 298.15 ± 1 K. Solutes and solvents are simply identified by numbers, see Table D.1 in Appendix D. The color code denotes the values of D_{ij}^{∞}, and white cells indicate missing entries.

To the best of the author's knowledge, this database is the first comprehensive database of diffusion coefficients at infinite dilution. However, of the 10,608 different possible combinations of the considered solutes and solvents, data are available only for 353 (3.3%). Furthermore, the resulting matrix is not only sparsely but also heterogeneously filled with observed entries, cf. Fig. 25; for instance, for the solvent water (column 28), a very large number of data points (with different solutes) is available, whereas many other solvents (and solutes) have been studied in only a very limited number of mixtures. In fact, a substantial share of the solutes that were studied in combination with water have not been studied in combination with any other solvent with regard to D_{ij}^{∞}.

4.4.3 Prediction of Diffusion Coefficients

4.4.3.1 Semiempirical Models

The experimental database shown in Fig. 25 was used for studying the performance of four established semiempirical models for the prediction of D_{ij}^{∞}, namely those of Wilke and Chang (1955) [94], Reddy and Doraiswamy (1967) [95], Tyn and Calus (1975) [96], and the Stokes-Einstein Gierer-Wirtz Estimation (SEGWE) by Evans at al. (2018) [97]. All considered models have in common that they predict D_{ij}^{∞} as a function of the quotient T/η_j, where T is the temperature in Kelvin and η_j is the dynamic viscosity of the solvent j. Hence, information of η_j at the temperature of interest is required. Furthermore, they all require information on the pure solute i, namely either the molar volume v_i, the molar mass M_i, or the parachor P_i - or some combination thereof. All pure-component properties were obtained in the present chapter from DIPPR correlations taken from the DIPPR database [113]. The Wilke-Chang and SEGWE models additionally require solvent-specific parameters. The authors provide some values of these parameters in the original publications, but in practice the parameters are typically first fitted to experimental data on D_{ij}^{∞} in the respective solvent j.

For the comparison of the semiempirical models with the MCMs, the solvent-specific parameters of Wilke-Chang and SEGWE have been fitted to data on D_{ij}^{∞} using a leave-one-out procedure (cf. Section 4.4.3.2.4). This procedure ensures a fair comparison between the semiempirical models and the MCMs. However, when SEGWE was used as prior information for the hybrid MCMs, the parameter was not fitted but instead a fixed global value was used. More information on the hybridization of SEGWE and an MCM is given in Sections 4.4.3.2.2-4.4.3.2.3.

Furthermore, details on the semiempirical models are provided in Appendix D.1.

4.4.3.2 Matrix Completion Methods

Three different MCMs were developed and evaluated in the present chapter: one MCM that is purely data-driven, i.e., which is only trained on the available experimental data for D_{ij}^{∞}. Furthermore, two hybrid MCMs, which additionally incorporate information from the SEGWE model in different ways as described below. All MCMs follow a Bayesian approach, cf. Chapter 2.

While different priors were chosen in the different MCMs, the same likelihood in form of a Cauchy distribution with scale $\lambda = 0.2$ was chosen for all MCMs. Both the form of the prior and the likelihood, including the scale parameter λ, are hyperparameters of the model. In preliminary studies with different configurations, the hyperparameter set from Chapter 4.1 proved to be most suitable, which was therefore adopted here. All feature vectors are of length K, where K is the number of features considered for each solute and each solvent. K is a further hyperparameter of the model and is a priori unknown; it must be chosen so that over- and underfitting are avoided. In preliminary studies, $K = 2$ was found the most suitable choice and was therefore used for all models here.

Gaussian mean-field variational inference has been employed using the Automatic Differentiation Variational Inference (ADVI) [19] option implemented in the probabilistic programming framework Stan [20], which was used for training all models. The code is attached in Appendix D.6.

4.4.3.2.1 Data-Driven Matrix Completion Method

The training of the purely data-driven MCM is based only on uncovering structure in the sparse matrix of experimental D_{ij}^{∞}. Each $\ln D_{ij}^{\infty}$ is thereby modeled as the dot product of the two latent feature vectors $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$:

$$\ln D_{ij}^{\infty} = \boldsymbol{u}_i \cdot \boldsymbol{v}_j + \varepsilon_{ij} \tag{36}$$

Here, $\ln D_{ij}^{\infty} \equiv \ln\left(\frac{10^9 D_{ij}^{\infty}}{\mathrm{m^2/s}}\right)$ is defined as the natural logarithm of the numerical value of the diffusion coefficient in 10^{-9} $\mathrm{m^2/s}$, which is used for scaling purposes.

During the training of the MCM, the generative model first draws two vectors $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$ of length K with features for each solute i and solvent j from the prior, for which a normal distribution centered around zero with a standard deviation $\sigma_0 = 1$ was chosen here. It then models the probability of each experimental data point $\ln D_{ij}^{\infty}$ as a Cauchy distribution with scale λ centered around the dot product of the respective feature vectors, cf. Eq. (36), and thereby adjusts the features so that they are best suited

for describing the training data, i.e., minimizing the ε_{ij}. When performing Bayesian inference, the probabilistic model is thereby inverted to obtain the posterior, i.e., the probability distribution over the features after considering the training data. The final features of the solutes and solvents were then obtained by taking the mean of the posterior, which was subsequently used for calculating predictions for $\ln D_{ij}^{\infty}$ with Eq. (36) (while setting ε_{ij} to zero).

4.4.3.2.2 Hybrid Matrix Completion Method "Boosting"

This MCM combines information from the experimental data on D_{ij}^{∞} with information from SEGWE and is thereby based on the concept of *Boosting* [114]. The idea of this hybrid approach is to train an MCM not on the experimental data for D_{ij}^{∞} (or $\ln D_{ij}^{\infty}$), but on the *residuals* res_{ij} of the SEGWE model:

$$\mathrm{res}_{ij} = \ln D_{ij}^{\infty,\mathrm{SEGWE}} - \ln D_{ij}^{\infty,\mathrm{exp}} = \boldsymbol{u}_i \cdot \boldsymbol{v}_j + \varepsilon_{ij} \tag{37}$$

Hence, in this case, the MCM is not employed to uncover structure in the experimental data, but in the *deviations* of the SEGWE predictions from the experimental data.

For the Boosting approach, SEGWE was applied in a purely predictive manner; this means that the parameter ϱ_{eff}, cf. Eq. (D.5), was not treated as a fit parameter but globally set to the value $\varrho_{\mathrm{eff}} = 619\ \mathrm{kg/m^3}$ as suggested by the original authors [97].

SEGWE has been chosen for the Boosting approach for two reasons: first, SEGWE proved to be the best-performing of the studied semiempirical models, cf. Section 4.4.4.1. Second, in the chosen variant of SEGWE, the only component descriptors required in the model equation are the viscosity of the solvent and the molar masses of solute and solvent; information on these properties is readily available.

The training of this hybrid MCM was carried out analogously to the data-driven approach, and with the same hyperparameters (prior and likelihood as well as number of features per solute / solvent K). After the training, MCM-Boosting yields predictions of the residuals of the SEGWE model for specified mixtures i, j. The respective predicted $\ln D_{ij}^{\infty}$ (and thus D_{ij}^{∞}) can then be calculated from the predicted residuals by rearranging Eq. (37).

4.4.3.2.3 Hybrid Matrix Completion Method "Whisky"

Furthermore, a second hybrid MCM for the prediction of D_{ij}^{∞}, which also combines information from experimentally available D_{ij}^{∞} with information from SEGWE, was

considered here. In contrast to MCM-Boosting, this hybrid model does not operate on the residuals of SEGWE, but is trained in two subsequent steps on two different data sets. The approach can be considered as a form of *distillation* of a model, which is why it is labeled MCM-*Whisky*. The approach is similar to the one recently introduced for the prediction of activity coefficients [9]. Therefore, only a brief description is given here, and the original work is referred to for an in-depth discussion.

The training of the Whisky model consists of two steps. In the first training step, the predictions of $\ln D_{ij}^{\infty}$ obtained with SEGWE (again with globally fixed $\varrho_{\text{eff}} = 619$ kg/m^3) for all combinations of the considered solutes and solvents were used for training a data-driven MCM according to Eq. (36) (while again using the same hyperparameters as in the MCMs described above). As result, *preliminary* feature vectors $\boldsymbol{u}_i^*$ and $\boldsymbol{v}_j^*$ of the solutes i and solvents j, respectively, were obtained. This training step can be interpreted as *distilling* the essence of the SEGWE model and storing it in the preliminary feature vectors $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$; therefore, this first training step is referred to as the *distillation step* in the following.

In the second training step, the preliminary feature vectors $\boldsymbol{u}_i^*$ and $\boldsymbol{v}_j^*$ were refined using the (sparse) experimental data on D_{ij}^{∞} from the database; the second training step is therefore referred to as the *maturation step* in the following. In the maturation step, the preliminary $\boldsymbol{u}_i^*$ and $\boldsymbol{v}_j^*$ were used for creating an *informed* prior for the training of an additional MCM, which was then trained on the experimental D_{ij}^{∞}. Specifically, the means of the respective preliminary features were adopted, whereas the standard deviations of the features were scaled with a constant factor, such that the mean of all resulting standard deviations was $\sigma = 0.5$.

This scaling procedure was carried out analogously to the approach described in Chapter 4.1 and ensures that the model remains flexible enough to reasonably consider the experimental training data. The final informative prior for the maturation step of the hybrid MCM was then obtained by multiplying the scaled posterior from the distillation step with the uninformed prior distribution as used in the data-driven MCM. This last step ensures that the informed prior is in all cases stronger than the uninformed prior.

Hence, in this hybrid MCM, information from SEGWE is included and transferred via the prior in the maturation step. However, the model is still capable of overruling the prior information from SEGWE via the likelihood, if the available experimental data for D_{ij}^{∞} is convincing enough to do so.

In both training steps of the Whisky model, the same likelihood (Cauchy with scale parameter $\lambda = 0.2$) and the same number of features per solute and solvent ($K = 2$) as in the other MCMs were used.

While both hybrid approaches, MCM-Boosting and MCM-Whisky, incorporate information from the SEGWE model, the difference is how the knowledge from the semiempirical model is encoded in the MCM as described above. MCM-Boosting can only lead to improvements over the baseline model (here: SEGWE) if that model shows *systematic* prediction errors. Only then can the MCM reveal structure in the residuals of the model and thereby refine the predictions. Furthermore, any information from SEGWE for mixtures for which no experimental data is available is inevitably discarded in the Boosting approach. In the Whisky approach, in contrast, different classes of training data are combined: predictions with the SEGWE model, which can be obtained for many mixtures (for the present data set, they could be obtained for all combinations of solutes and solvents) but are rather uncertain, and experimental data, which are rare (see Section 4.4.2) but more reliable than model predictions. For components for which there are many experimental data, the Whisky approach can be expected to hardly improve the predictive performance compared to a data-driven MCM. On the other hand, for components for which there are only few experimental data for training, the largest improvements compared to the data-driven MCM can be expected with the Whisky approach.

4.4.3.2.4 Leave-One-Out Analysis and Reduced Database

The predictive performance of all MCMs developed in this chapter was evaluated by a leave-one-out analysis [115]. Following this concept, each MCM was trained on a subset of the experimental data on D_{ij}^{∞} that includes all observed entries *except for the one to be predicted*. The single left-out data point, which is called *test data point* in the following, was then predicted by the MCM. This procedure was repeated by subsequently defining all data points once as test data point, until true predictions for all available D_{ij}^{∞} were obtained. Finally, these predictions were compared to the respective experimental D_{ij}^{∞} to evaluate the performance of the MCMs.

By nature, such a leave-one-out analysis of an MCM demands a database in which at least *two* distinct data points are available for each solute i and each solvent j, so that after declaring one of these data points as test data point, there is at least one data point for each component in the training set to allow the model to learn its characteristics. Hence, if the database is arranged in matrix form with solutes and solvents representing the rows and columns, respectively, at least two observed entries per row and per column are required for a meaningful analysis.

Therefore, for developing the MCMs, a *reduced database* for D_{ij}^{∞} that satisfies the aforementioned condition was defined. To enable a direct comparison, the predictive performance of the semiempirical models was also evaluated based on this reduced data

set. Thereby, the solvent-specific parameters of the models of Wilke and Chang and SEGWE were also fitted to experimental data for D_{ij}^{∞} in a leave-one-out approach (cf. Appendix D.1.5).

The reduced database is presented in Fig. 26. It is the basis for the comparison of the performance of the three MCMs and the semiempirical models for predicting D_{ij}^{∞} considered in the present chapter.

While the MCM only works for mixtures within the matrix shown in Fig. 26, the semiempirical models can also give predictions for additional mixtures outside the matrix, namely for all mixtures for which the required pure-component properties are known.

The reduced database comprises data for 45 solutes and 23 solvents. The corresponding matrix, which is shown in Fig. 26, has about 16% observed entries: for 166 of the 1035 possible mixtures experimental data are available.

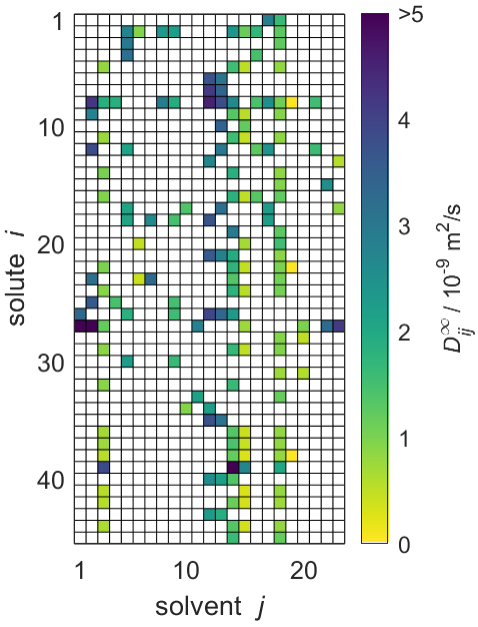


Figure 26: Overview of the experimental data for the liquid-phase diffusion coefficients D_{ij}^{∞} at infinite dilution at 298.15 ± 1 K in the reduced database; these data points were used for evaluation of the MCMs developed in the present chapter and comparison of the results to those of the semiempirical models. Solutes and solvents are identified by numbers, see Tables D.2 and D.3 in Appendix D. The color code indicates the value of D_{ij}^{∞}, and white cells denote missing data.

Four particularly well-filled columns can be discerned for $j = 3, 14, 15$, and 18. The respective solvents are ethanol, methanol, *n*-propanol, and water. They are common solvents for which experimental data were measured in combination with many solutes. Moreover, a column-based structure can be observed in the absolute values of D_{ij}^{∞} themselves (and not just in the availability of data): for example, the diffusion coefficients

in the solvent methanol (j = 14) are consistently higher than the respective diffusion coefficients in the solvent n-propanol (j = 15), which is readily seen by the darker colors in that column in Fig. 26. Two further solvents, n-hexane and n-heptane (j = 12 and j = 13, respectively), exhibit even darker colors, corresponding to even higher values of D_{ij}^{∞}. Similar structural relationships in the matrix exist also for the rows, e.g. for carbon dioxide (i = 39) comparatively large diffusion coefficients are found. It will be shown below that the MCMs developed in the present chapter are able to pick up on these relationships and even identify more complex relationships in the data structure, which are veiled before the human eye.

The predictive performance of the methods was analyzed and compared in terms of a relative mean absolute error (rMAE), cf. Eq. (38), and a relative root mean squared error (rRMSE), cf. Eq. (39), which were calculated by comparing the predictions (pred) obtained during the leave-one-out analysis to the respective experimental data (exp):

$$\mathrm{rMAE} = \frac{1}{N} \sum_i \sum_j \left| \frac{D_{ij}^{\infty,\mathrm{pred}} - D_{ij}^{\infty,\mathrm{exp}}}{D_{ij}^{\infty,\mathrm{exp}}} \right| \tag{38}$$

$$\mathrm{rRMSE} = \sqrt{\frac{1}{N} \sum_i \sum_j \left(\frac{D_{ij}^{\infty,\mathrm{pred}} - D_{ij}^{\infty,\mathrm{exp}}}{D_{ij}^{\infty,\mathrm{exp}}} \right)^2} \tag{39}$$

where N is the total number of available experimental data points for D_{ij}^{∞} in the reduced database and the summation is iterated over all considered solutes i and solvents j.

4.4.4 Results and Discussion

In Fig. 27, the performance of the four studied semiempirical models, as well as that of the three developed MCMs for the prediction of the D_{ij}^{∞} from the reduced database, are compared in terms of the relative mean absolute error (rMAE) and the relative root mean squared error (rRMSE), cf. Section 4.4.3.2.4.

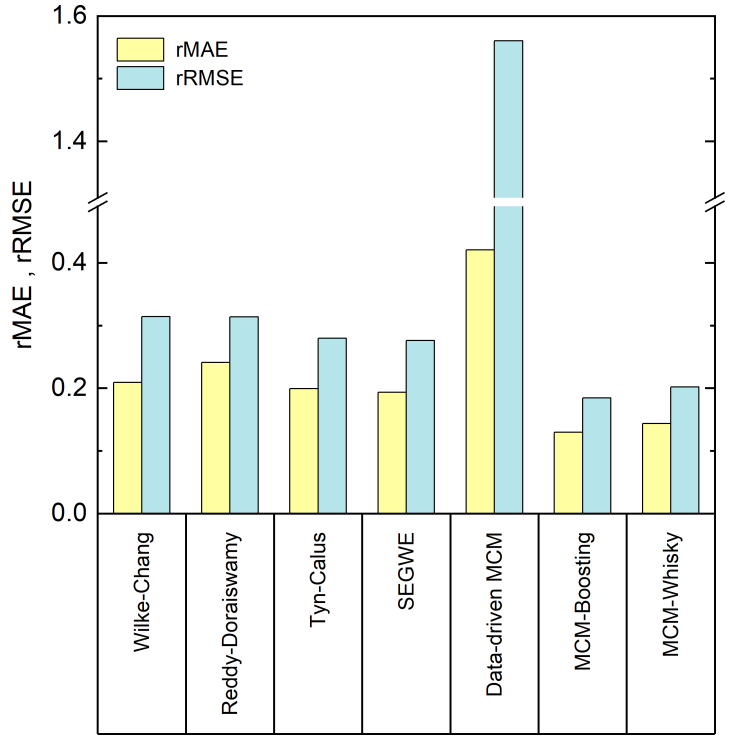


Figure 27: Relative mean absolute error (rMAE, yellow) and relative root mean squared error (rRMSE, blue) of the predicted D_{ij}^{∞} with the studied semiempirical models and the developed MCMs for the experimental data from the reduced database.

4.4.4.1 Prediction of D_{ij}^{∞} with Semiempirical Models

Let us first compare the results of the four semiempirical models.

A similar performance of all semiempirical models in both error metrics is observed. The rMAE is about 0.20, and below 0.25 in all cases, with the largest value (poorest performance) found for the model of Reddy and Doraiswamy and the lowest value (best performance) found for SEGWE. Also, the values for the rRMSE vary only slightly between the different models and range from 0.31 (Reddy-Doraiswamy) to 0.28 (SEGWE). Although the four semiempirical models do not vary substantially in their rRMSE scores, a continuously decreasing rRMSE with the year of publication of the respective model can be observed. It can be speculated that this is an effect of the increasing availability of experimental data to which these models were fitted.

It is also important to note that, at the time these works were published, the authors presumably used the entirety of available data on D_{ij}^{∞} for developing their models. This means that substantial parts of the data on which their performance is evaluated have already been *seen* by the semiempirical models.

Comparing the rMAE and the rRMSE from the semiempirical models directly with the corresponding values from the MCMs, as it is done in Fig. 27, therefore creates a bias,

which favors the semiempirical models; the calculation of the rMAE and RMSE for the MCMs, in contrast, is based on a strict application of the leave-one-out strategy, i.e., none of the predicted values was part of the training set, which is not the case for the development of the semiempirical models. The fact that the fitting of solute-specific model parameters (of Wilke-Chang and SEGWE) was carried out with a leave-one-out technique does not change the above statement, as the model development was nonetheless based on all available data at that time.

Overall, SEGWE shows the best performance of the studied semiempirical models in both rMAE and rRMSE, and was therefore considered as benchmark against which the MCMs developed in the present chapter are compared in the following.

4.4.4.2 Prediction of D_{ij}^{∞} with Matrix Completion Methods

The performance of the methods for the prediction of D_{ij}^{∞} developed in this chapter is discussed in the following. These methods include the purely data-driven MCM and the hybrid approaches based on Boosting, which is called *MCM-Boosting*, and the one based on model distillation, which is called *MCM-Whisky*.

The rMAE and rRMSE scores of the data-driven MCM are 0.42 and 1.56, respectively, which is much higher than those of all studied semiempirical models, cf. Fig. 27. The data-driven MCM thereby strongly suffers from a poor prediction of D_{ij}^{∞} in mixtures with the solvent 1,2-propanediol; namely the D_{ij}^{∞} in the mixtures (benzene + 1,2-propanediol) and (1,3-dihydroxybenzene + 1,2-propanediol) are predicted with extremely large relative errors of 1,397% and 1,339%, respectively, which results in a large rMAE and a particularly large rRMSE score for the data-driven MCM. As shown in Fig. 26, the experimental D_{ij}^{∞} for the solvent 1,2-propanediol (j = 19) are extremely small, namely about 2 orders of magnitude lower than the bulk of the data. Hence, already small *absolute* deviations between prediction and experimental D_{ij}^{∞} lead to extremely large errors on the *relative* scale, i.e., large values of rMAE and rRMSE, here. Excluding just the two mentioned data points from the evaluation improves the score of the data-driven MCM to 0.26 (vs 0.42 with the points included) in the rMAE and 0.42 (vs 1.56 with the points included) in the rRMSE - still slightly worse, but in the same range as the performance of the semiempirical methods.

An important requirement for the success of data-driven prediction methods in general, and the introduced data-driven MCM here in particular, is the availability of training data. One way to evaluate the data situation is comparing the number of available data points for training the model to the number of model parameters, which, among others, depends on the number of different components considered by the model. Therefore, an observation ratio $r_{\mathrm{obs}} = \frac{N_{\mathrm{obs}}}{m+n}$ can be assessed, as done in Ref. [10], where N_{obs} is

the number of observed entries of the sparsely populated matrix and m and n are the numbers of rows and columns of the matrix, i.e., considered solutes and solvents, respectively.

In Ref. [10], a strong correlation of the predictive performance of MCMs for the prediction of activity coefficients at infinite dilution was found with r_{obs}, which was between 4.4 and 9.2 in that study [10]. Rather high values of r_{obs} led to a significantly better performance than rather low values. In the present study, the value of r_{obs} is 2.4, which is substantially smaller than the lowest studied value in Ref. [10]. This indicates that the situation regarding availability of training data is highly challenging here, in particular for the data-driven MCM, which leaves ample room for improvements. It is noted here that other factors besides the mere number of training data points are also important, such as the heterogeneity in the number of available data for different components.

Such improvements can, as shown in Fig. 27, be achieved by hybridizing the data-driven MCM with information from SEGWE: both hybrid MCMs perform significantly better than all established semiempirical models and the data-driven MCM in both error scores rMAE and rRMSE. Let us first discuss the results of MCM-Boosting.

The key idea of MCM-Boosting is to train the algorithm on the residuals of the SEGWE model, and not on experimental data directly, cf. Section 4.4.3.2.2. In Fig. 28, the residuals between the SEGWE predictions and the data from the reduced database, cf. Eq. (37), are plotted. Here, SEGWE was applied in the purely predictive variant with a globally fixed $\varrho = 619$ kg/m^3 to ensure that no information on the test data point was included in the training of MCM-Boosting. Fig. 28 basically shows the performance of SEGWE for each individual data point from the reduced database. Large deviations are observed, indicated by the color code in Fig. 28, in particular for the solutes water ($i = 27$) and carbon dioxide ($i = 39$), but beyond that no apparent structure in the residuals is immediately recognizable. A more detailed discussion of the mixtures for which SEGWE gives predictions with particularly large errors is included in Appendix D.1.6.

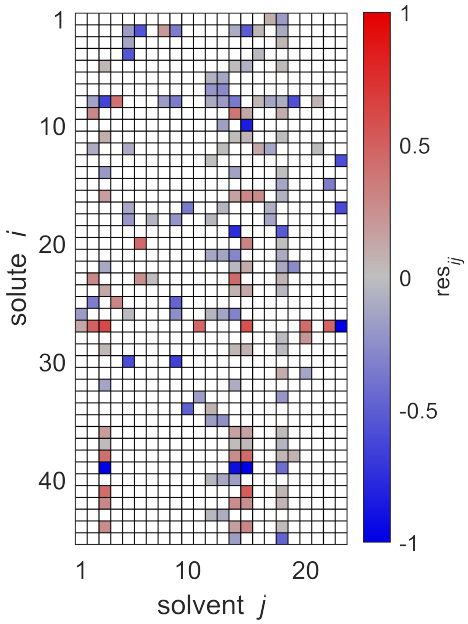


Figure 28: Residuals res_{ij} of the SEGWE predictions from the experimental data for D_{ij}^{∞} at 298.15 ± 1 K from the reduced database. Solutes i and solvents j are identified by numbers, see Tables D.2 and D.3 in Appendix D. The color code indicates the value of res_{ij}, and white cells denote missing data.

The diffusion coefficients predicted by MCM-Boosting show overall a very good agreement with the literature values. The rMAE and rRMSE (cf. Fig. 27) are 0.130 and 0.184, respectively. The performance of MCM-Boosting is not just better in the averaged scores: as shown in Fig. D.3 of Appendix D, the maximum prediction error found for any mixture is lower for MCM-Boosting than for all other investigated methods.

The second hybrid model, MCM-Whisky, which uses - besides information from the experimental training data – information from SEGWE via an informed prior, cf. Section 4.4.3.2.3, also performs significantly better than the data-driven MCM and all semiempirical models. The rMAE and rRMSE of MCM-Whisky are 0.143 and 0.202, respectively, cf. Fig. 27, making the overall performance close to but slightly worse than that of MCM-Boosting.

For an improved evaluation of the results of the hybrid MCMs, the respective predictions for D_{ij}^{∞} are additionally shown in parity plots over the experimental data from the reduced database in Fig. 29. For comparison, a parity plot showing the predictions of the best semiempirical model, namely SEGWE with a solvent-specific fitted ϱ_{eff} (cf. Section D.1.4), is also included in Fig. 29.

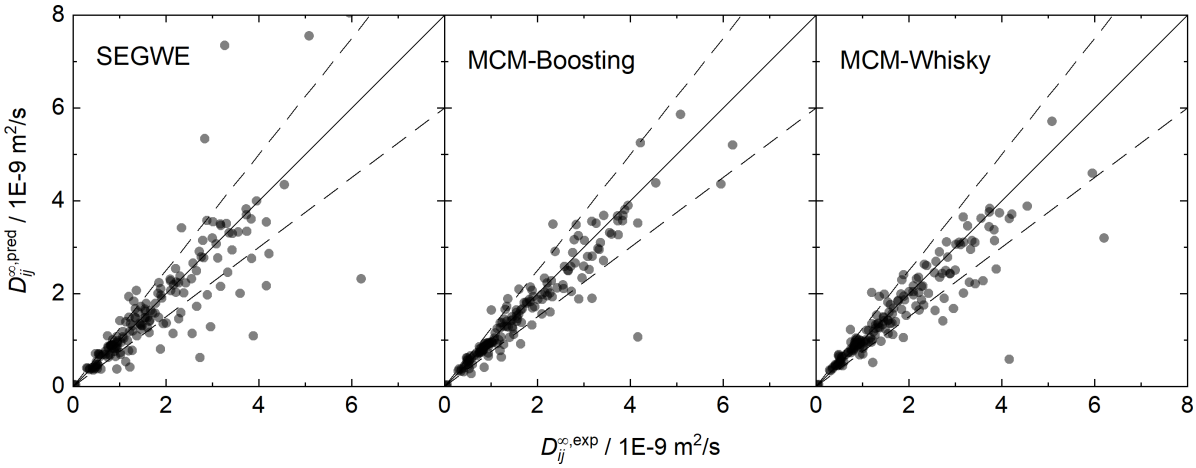


Figure 29: Parity plots of the predictions (pred) of D_{ij}^{∞} with SEGWE and both hybrid MCMs developed in this chapter over the experimental data (exp) from the reduced database. The solid lines indicate perfect predictions, the dashed lines indicate relative deviations of ±25%.

The parity plots for the two hybrid MCMs show a narrow spread of the data points around perfect predictions (solid lines), and in general only few outliers that are predicted with very large deviation; most of the predicted data points lie within the ±25% boundaries (dashed lines). Slightly more data points are underestimated by MCM-Whisky compared to MCM-Boosting, which is the reason for the slightly higher rMAE and rRMSE scores. In contrast, SEGWE shows a comparatively large number of predictions outside the ±25% boundaries.

The results of MCM-Boosting (the overall best-performing MCM) are also compared to those of SEGWE (the overall best-performing semiempirical model) in a histogram representation in Fig. 30, which shows the number of data points that are predicted with a certain relative deviation from the experimental data.

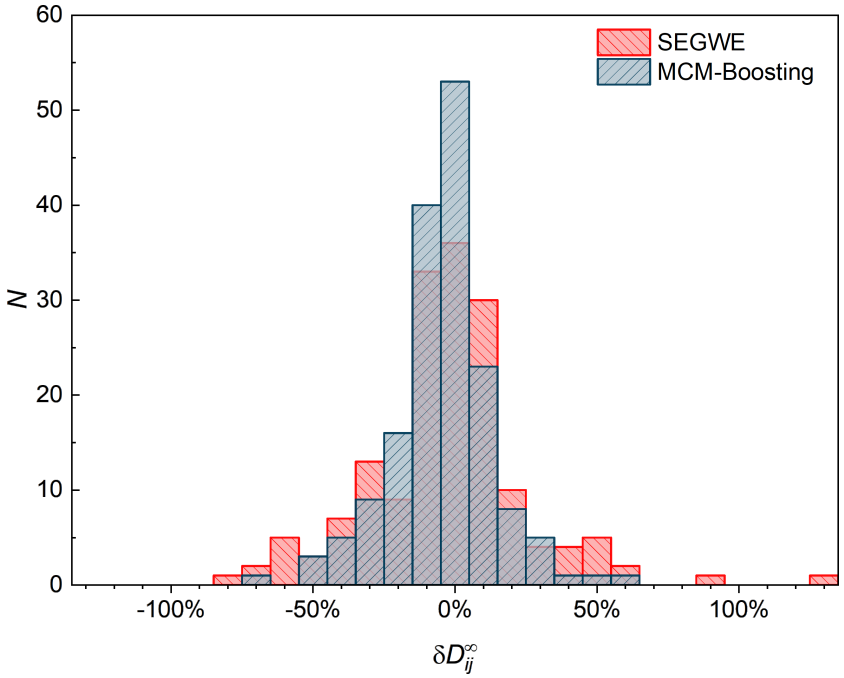


Figure 30: Histogram of the number of data points N from the reduced database that are predicted with a defined relative deviation from the respective experimental data $\delta D_{ij}^{\infty} = \left(D_{ij}^{\infty,\mathrm{pred}} - D_{ij}^{\infty,\mathrm{exp}}\right)/D_{ij}^{\infty,\mathrm{exp}}$ by SEGWE (red) and MCM-Boosting (blue).

Fig. 30 underpins the performance of the hybrid MCM-Boosting: more D_{ij}^{∞} are predicted with low deviation compared to the predictions by SEGWE. For instance, 116 data points are predicted with a relative error $\left|\delta D_{ij}^{\infty}\right| < 15\%$ with MCM-Boosting, whereas for SEGWE, this is the case for only 99 data points. The differences are even clearer when looking at predictions with a relative error $\left|\delta D_{ij}^{\infty}\right| < 5\%$: MCM-Boosting predicts 53 mixtures with such high accuracy, versus just 36 in the case of SEGWE.

4.4.4.3 Completed Database

As a final result, the completed matrices of D_{ij}^{∞} predictions using MCM-Boosting and MCM-Whisky are provided for the 10,608 possible combinations of all 208 solutes i and 51 solvents j from the full database, as introduced in Section 4.4.2. In this case, the MCMs have not been trained following a leave-one-out strategy, but using all data points from the database; the same hyperparameters were thereby used as in the previously described analysis. The complete predicted data set is provided in Ref. [83] in a machine-readable format, namely as a .csv file, together with the learned feature vectors $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$, from which the data can also be constructed. If predictions for unstudied D_{ij}^{∞} are required, they can be taken from this table.

For MCM-Boosting, the completed matrix of D_{ij}^{∞} predictions is visualized in Fig. 31, together with the uncertainties of the predictions. The corresponding visualization for MCM-Whisky is in Fig. D.4 of Appendix D.

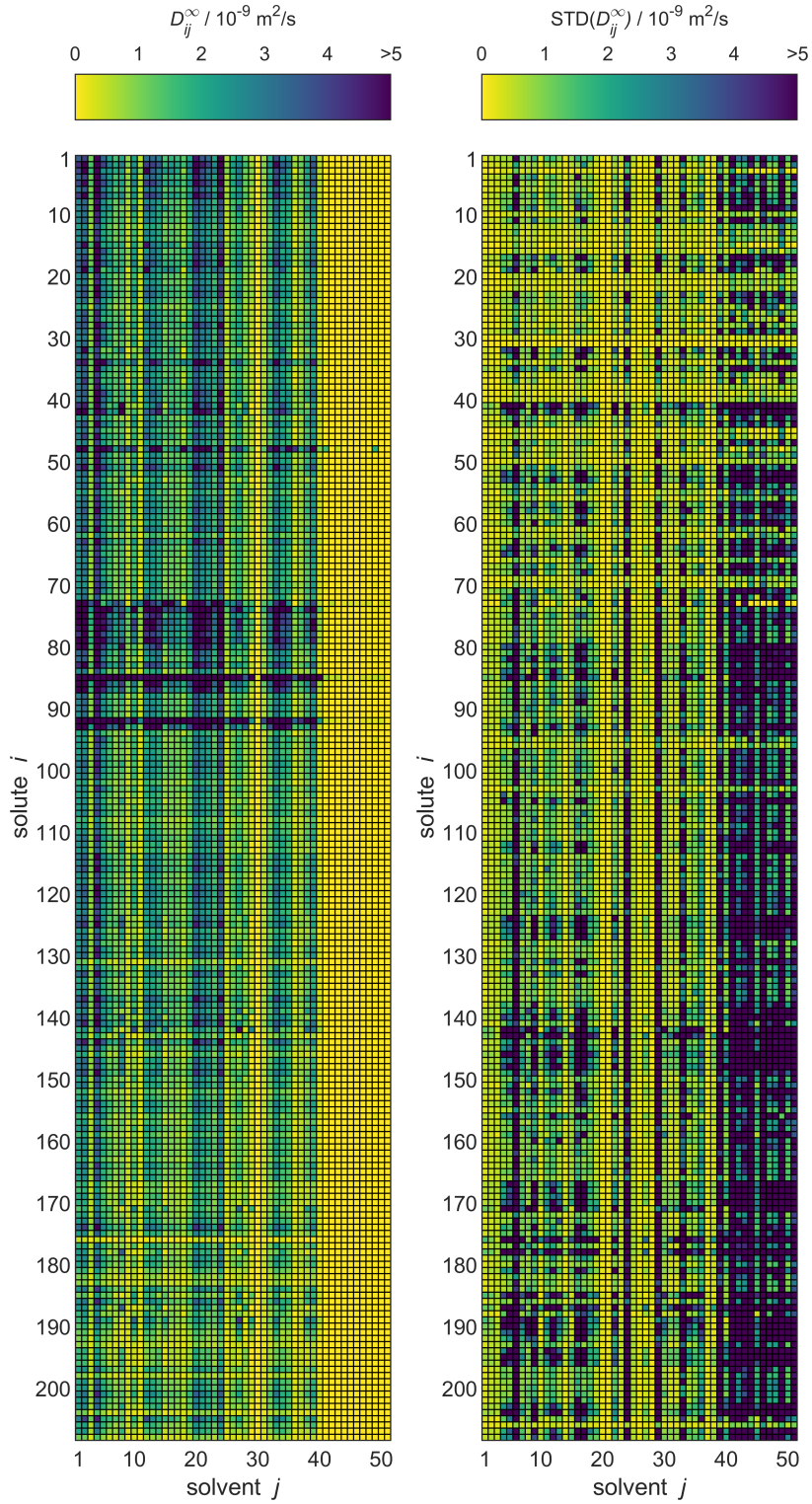


Figure 31: Predictions of D_{ij}^{∞} by MCM-Boosting (left) and the uncertainties of the predictions (right) for all solutes i and solvents j (identified by numbers, see Table D.1 in Appendix D) from the full database. The color code indicates the values of D_{ij}^{∞}.

A significant advantage of the Bayesian approach of matrix completion, which has been followed here, is that probability distributions for all predicted D_{ij}^{∞} with the MCMs are obtained. This allows the reporting of not only the predictions for D_{ij}^{∞}, but also the corresponding uncertainties. That information is also provided both for MCM-Boosting and MCM-Whisky in the .csv files in Ref. [83].

The methods presented in this chapter were applied here only to a single isotherm. The semiempirical models, on the other hand, describe diffusion data at arbitrary temperatures. In principle, the studies done in this chapter could be extended to include the influence of the temperature on D_{ij}^{∞}, as it was done by Damay et al. for the prediction of activity coefficients at infinite dilution [10]. However, such an endeavour is likely to encounter problems as the database on D_{ij}^{∞} is extremely narrow outside the range of ambient temperatures [83]. To achieve substantial advances, more data are needed, and in particular more data that cover a wider temperature range.

4.4.5 Conclusions

In the present chapter, a comprehensive database of liquid-phase diffusion coefficients at infinite dilution D_{ij}^{∞} in binary mixtures at 298.15±1 K was used. The database contains 353 experimental data points for D_{ij}^{∞} and covers 208 solutes i and 51 solvents j. It has been used for systematically evaluating four established semiempirical models for predicting D_{ij}^{∞}, namely the methods of Wilke and Chang, Reddy and Doraiswamy, Tyn and Calus, and SEGWE; the best performance was found for the most recent of these models, which is SEGWE.

Furthermore, novel methods have been developed for the prediction of D_{ij}^{∞} based on the machine learning concept of matrix completion. Three such matrix completion methods (MCMs) are presented here: a purely data-driven MCM, which was trained only on the data on the experimental D_{ij}^{∞}, and two hybrid MCMs that combine information from SEGWE with the experimental data. The purely data-driven MCM suffers from the sparsity of the available data and performs not as well as the semiempirical models. This is different for the two hybrid MCMs, for which significant improvements in terms of predictive accuracy compared to all semiempirical models were found.

5 Matrix Factorization of Group Interactions

5.1 Training on Group-Interaction Parameters: UNIFAC 1.1

5.1.1 Introduction

Methods for predicting thermodynamic properties are of paramount importance in chemical engineering, simply because there are too many relevant substances to study them all in experiments. The scale of this problem soars when going from pure components to mixtures, for simple combinatorial reasons. Also methodologically, predicting properties of mixtures is a demanding task. It can be tackled basically from two sides: on the one hand, one can look for similarities between substances (which is basically a data-driven approach), on the other hand, one can try to base predictions on physical theory.

The most successful methods in the field combine these two aspects. Among these, methods that rely on the concept of *group-contributions* (GC) play an important role. They are based on the idea that components can be characterized by the *structural groups* they contain and take advantage of the fact that the number of relevant structural groups is many orders of magnitude smaller than the number of relevant components. As a consequence, GC methods can be used for describing a very large number of components based on a relatively small number of *group-specific parameters*: any component that can be built from groups, for which parameters are available, can be modeled.

Basically all thermodynamic models of mixtures rely on describing *pair interactions*. Component-specific models, like UNIQUAC [116, 117] or NRTL [118], thereby describe the pairwise interactions between components using *component-specific* pair-interaction parameters, which need to be fitted to experimental data. Usually, data for binary mixtures are used for this purpose, which means that for modeling multi-component mixtures, binary mixture data are needed for all binary subsystems of the studied mixture. Unfortunately, due to the combinatorial problem, even data for binary mixtures

are often missing, which strongly limits the applicability of the component-specific models.

GC methods circumvent this problem. By dividing components into structural groups, GC methods only rely on *group-specific* pair-interaction parameters, namely *group-interaction* parameters, which are fitted to experimental mixture data, whereby the amount of required training data compared to component-specific models is significantly reduced.

One of the most successful thermodynamic group-contribution method for mixtures is UNIFAC, which was first introduced in 1975 [41] and has been significantly extended and refined since then [12, 119–123]. Also, several tailored versions of UNIFAC fitted for specific applications are available [124–126]. And there is also a commercial version of UNIFAC, provided within the UNIFAC-Consortium, which is based on the same model equations as the public versions of UNIFAC, but whose parameter tables have been revised and extended on a regular basis since 1996 [127] using both public data and non-public data provided or generated within the consortium. The scope of the commercial version is therefore larger than that of the public versions of UNIFAC. Since the commercial version is not freely accessible, the focus is placed here on the most recent public version of UNIFAC [12], which is referred to simply as UNIFAC in the following for brevity. The author has also access to the commercial version of UNIFAC, called UNIFAC-TUC in the following, but this version is used for comparisons only.

UNIFAC was derived from the component-specific lattice model UNIQUAC [116, 117] and describes the molar excess Gibbs energy g^{E} of a mixture as a function of temperature T and composition $\boldsymbol{x}$. Both energetic and entropic contributions to g^{E} are considered in the model. All versions of UNIFAC use geometric parameters for the individual structural groups, which describe their volume and surface and determine the entropic contribution. Furthermore, parameters describing the pairwise energetic interactions between the different structural groups in the mixture are used. These group-interaction parameters play the central role in the model.

From the excess Gibbs energy g^{E}, many properties that are essential in chemical engineering can be determined, most importantly the activity coefficients γ_i of the components i in the mixture, based on which phase equilibria can be predicted [41]. Over the years, many structural groups have been included in the UNIFAC parameter tables, so that a huge number of components of practical interest can be modeled. UNIFAC presently considers 54 *main groups*, which are further divided into 113 *subgroups* [12]. The difference between main groups and subgroups is that each subgroup k has individual geometric parameters, namely the group volume R_k and group surface area Q_k [128], while all subgroups that belong to the same main group m share the same group-interaction parameters. There are two distinct group-interaction parameters for

each binary combination of different main groups (m, n); they are generally labeled as a_{mn} and a_{nm}, and have, as a result of the fit, usually different values, i.e., $a_{mn} \neq a_{nm}$.

While Q_k and R_k are reported for 113 individual subgroups, there are still significant gaps regarding the group-interaction parameters a_{mn} and a_{nm} between the 54 main groups: there are 1,431 distinct binary combinations of unlike main groups ($m \neq n$), for which only for 635 (44%) group-interaction parameters have been reported yet. Fig. 32 schematically shows the publicly available set of group-interaction parameters between the first 50 main groups of UNIFAC [12]. The first 50 main groups were chosen here since for all of these, group-interaction parameters with at least five other main groups are publicly available to date. This threshold was chosen since, as described in detail below, the missing group-interaction parameters were predicted based on information from the *available* parameters only. For the sake of completeness, Fig. E.1 in Appendix E shows for which of the group combinations parameters are available in the commercial UNIFAC-TUC.

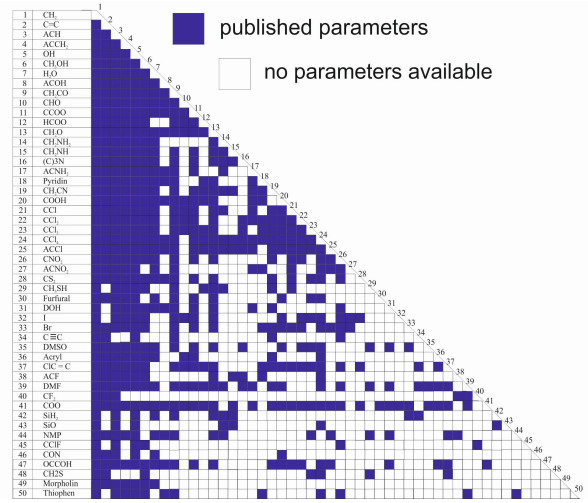


Figure 32: Matrix representing the availability of group-interaction parameters of UNIFAC [12] up to main group 50. Blue: parameters available.

Hence, the availability of the parameters describing the individual groups R_k and Q_k generally poses no problem, whereas missing group-interaction parameters a_{mn} and a_{nm} significantly limit the applicability of all versions of UNIFAC. The main reason why these gaps still persist, after so many years of work on the development of UNIFAC, is that the database for their determination is simply too narrow. There are structural groups that occur in many molecules, such as the methyl group or the hydroxyl group, and there are less common groups. It is particularly these less common groups for which the parameters are lacking. This is not to say that these groups do not occur in interesting components, but there are simply less data on binary mixtures containing components

with these groups. It is evident that this causes problems in the parameterization of UNIFAC.

A further drawback is that fitting group-interaction parameters is still not a routine, but rather artwork, in particular regarding the selection of the considered data sets, including their initial evaluation and consistency checking, and regarding the selection of a suitable objective function to be minimized during the fitting procedure. For a more detailed description of the fitting procedure of UNIFAC group-interaction parameters, it is referred to the literature[41, 129–131].

In this chapter, a method for the *prediction of the complete set* of the group-interaction parameters of group-contribution methods based on an existing parameter set is presented, without requiring new experimental data. The basic idea is to consider the group-interaction parameters as entries of a squared matrix (which is only partially filled, as several parameters are missing), and to use a matrix completion method (MCM) [14, 132] to estimate the missing entries. To demonstrate the applicability of this approach, it is applied to UNIFAC [12], for which the complete set of the group-interaction parameters for the first 50 main groups is predicted. Fig. E.2 in Appendix E gives an overview of this approach.

Following an idea developed in a recent paper [84], in which an MCM has been applied for estimating the component-specific pair-interaction parameters of UNIQUAC, the *asymmetric* group-interaction parameters ($a_{mn} \neq a_{nm}$) are not used directly, but rather the *symmetric* group-interaction *energies* $U_{mn} = U_{nm}$. The parameters of the two types (a and U) are connected by:

$$\begin{aligned} a_{mn} &= U_{mn} - U_{nn} \\ a_{nm} &= U_{nm} - U_{mm} \end{aligned} \tag{40}$$

Hence, according to Eq. (40), a_{mn} and a_{nm} are not independent but correlated.[1] Despite this, for parameterizing UNIFAC, a_{mn} and a_{nm} are usually considered to be uncorrelated. The fitting then results in a parameter set that does not comply with Eq. (40), cf. Ref. [84]. The approach proposed in this chapter overcomes this inconsistency.

In a series of recent papers, the capabilities of MCMs for predicting different types of thermodynamic data of mixtures using various component-based approaches have been demonstrated [1, 8–10, 82, 84]. However, these component-based approaches are inherently limited regarding the number of components that are covered; the respective

[1] For an N-component mixture, there are $N^2 - N$ asymmetric pair-interaction parameters of the a-type (the diagonal remains empty or is filled with zeros), while there are $(N^2 - N)/2 + N$ symmetric pair-interaction energies of the U-type (the diagonal is occupied by the pure-component energies, but only one of the triangular matrices has to be filled due to the symmetry). It is always possible to determine the a-parameters from the U-parameters, but not vice versa.

models complete a matrix spanned by the components that are part of the mixtures in the training set. This is not the case for the group-contribution methods, which are considered in the present chapter: as the groups form building blocks from which components can be created flexibly, the scope of the group-contribution methods for mixture properties is inherently extremely large – and it can now be extended substantially by using an MCM to complete the set of group-interaction parameters.

The approach proposed here should also be applicable to any other version of UNIFAC and to other group-contribution models for predicting thermodynamic properties of mixtures that are based on pair interactions. One advantage of the approach is that it can be put into practice, for example, by being integrated into existing process simulators in a very simple and straightforward manner: the existing UNIFAC parameter set of the model implementation only needs to be replaced by the predicted one provided by the proposed approach. For other machine-learning approaches, like artificial neural networks operating on molecular graphs [48, 133] or SMILES representations of the components [49], this might be more complicated in practice.

5.1.2 Method

The applicability of using MCMs for the prediction of group-interaction parameters of thermodynamic group-contribution methods is demonstrated by applying it to UNIFAC [12]. The resulting new version of UNIFAC (in which the predicted new parameters are used) is called *UNIFAC-MCM* in the following.

The MCM that was used in the present chapter is based on Bayesian matrix factorization (cf. Chapter 2) and similar to the ones used in Refs. [1, 8, 9, 82, 84]. In principle, the MCM could have been applied directly to the matrix of the a-type parameters, i.e., the matrix containing the group-interaction parameters a_{mn} and a_{nm}. However, this option was discarded for the following reasons: firstly, the available values for a_{mn} and a_{nm} are inconsistent with Eq. (40). Also, fitting a_{mn} and a_{nm} to mixture data can give different combinations of these parameters yielding basically equivalent results for the physical properties to which they were fitted [134]. This hinders an interpretation of these parameters and makes them poor candidates for applying an MCM. These problems were overcome by working with the group-interaction *energies* U_{mn} as explained below. Furthermore, in applying the MCM to the a matrix, the target function would have been to achieve an optimal representation of the a-type parameters. However, with UNIFAC-MCM, an optimal description of activity coefficients is rather of interest than a representation of model parameters. UNIFAC-MCM was therefore trained on pseudo-data for activity coefficients as described in the next section.

5.1.2.1 Training Data

As training data for UNIFAC-MCM, pseudo-data have been generated for the logarithmic activity coefficients $\ln \gamma_{mn}$ in hypothetical binary mixtures of the "pure main groups" of UNIFAC (m and n) at different temperatures and group mole fractions. Here, $\ln \gamma_{mn}$ represents the logarithmic activity coefficient of m in the binary mixture with n. For any given temperature and mole fraction, there are two distinct values $\ln \gamma_{mn}$ and $\ln \gamma_{nm}$, respectively, which can be represented in a matrix. The diagonal elements of this matrix are occupied with ones by definition and were not considered here. For simplicity, $\ln \gamma_{mn}$ will simply be used in the following to refer to that matrix, which includes the values from both triangular matrices, $\ln \gamma_{mn}$ and $\ln \gamma_{nm}$.

Specifically, $\ln \gamma_{mn}$ have been calculated for all binary combinations of the first 50 main groups of UNIFAC for which the required parameters were available, which holds for 619 combinations (or 50.5% of all possible binary combinations of these main groups). The grid was spanned by $T \in \{250, 300, 350, 400, 450\}$ K for the temperature, which covers the temperature of most of the available experimental data, and $x_m \in \{0.01,0.2,0.4,0.6,0.8, 0.99\}$ mol/mol for the composition.

For generating the pseudo-data for $\ln \gamma_{mn}$, the UNIFAC equations (cf. Eqs. (E.1) - (E.11) in Appendix E) were used in the common manner for hypothetical components that were composed of a single main group in all cases. For main groups with several subgroups k (with individual geometric parameters Q_k and R_k), the values of Q_k and R_k for one of the respective subgroups were selected, for details see Table E.1 in Appendix E. In principle, UNIFAC-MCM could also be trained on data for the residual part of the activity coefficients alone, which describes the energetic interactions (cf. Eq. (E.7) in Appendix E), because the interaction parameters only occur in this term. Also, this option has been tested, and the results were found to be very similar to those reported here, as expected.

5.1.2.2 Matrix Factorization

At its heart, UNIFAC-MCM factorizes the matrix of group-interaction energies U_{mn} between UNIFAC main groups m and n. The *unlike* U_{mn} ($m \neq n$) are modeled as the sum of two dot products:

$$U_{mn} = U_{nm} = \boldsymbol{\theta}_m \cdot \boldsymbol{\beta}_n + \boldsymbol{\theta}_n \cdot \boldsymbol{\beta}_m \tag{41}$$

where $\boldsymbol{\theta}_m$ and $\boldsymbol{\beta}_m$ as well as $\boldsymbol{\theta}_n$ and $\boldsymbol{\beta}_n$ are vectors of length K containing a priori unknown (latent) features of the UNIFAC main groups m and n, respectively. $\boldsymbol{\theta}_m$, $\boldsymbol{\beta}_m$, $\boldsymbol{\theta}_n$, and $\boldsymbol{\beta}_n$ are parameters of UNIFAC-MCM, while K is a hyperparameter that

controls the number of features considered per main group and thereby determines the flexibility of the model. Based on results of Ref. [84], K was set to $K = 3$ here. The form of Eq. (41) was chosen to ensure that all resulting group-interaction energies are symmetric, as required by the lattice model. Besides the unlike interaction energies, also *like* group-interaction energies U_{mm} are needed, cf. Eq. (40). They were not included in the factorization (Eq. (41)) but determined directly in the fit.

For training UNIFAC-MCM on the pseudo-data for $\ln \gamma_{mn}$, cf. Section "Training Data", the matrix factorization of the group-interaction energies U_{mn}, cf. Eq. (41), as well as Eq. (40), which relates the U_{mn} to the group-interaction parameters a_{mn}, were embedded in the UNIFAC equations, cf. Eqs. (E.1) - (E.11) in Appendix E. This establishes a generative probabilistic model for the $\ln \gamma_{mn}$. The training data were hence modeled by:

$$\ln \gamma_{mn}(T, x_m) = \mathrm{UNIFAC}(T, x_m, \boldsymbol{\theta}_m, \boldsymbol{\theta}_n, \boldsymbol{\beta}_m, \boldsymbol{\beta}_n, U_{mm}, U_{nn}) + \varepsilon_{mn} \tag{42}$$

where ε_{mn} is the deviation between the modeled $\ln \gamma_{mn}$ and the training data. The model parameters $\boldsymbol{\theta}_m$, $\boldsymbol{\theta}_n$, $\boldsymbol{\beta}_m$, $\boldsymbol{\beta}_n$, U_{mm}, and U_{nn} were fitted in a Bayesian framework to minimize these deviations. For more details on the implementation of the model and the training procedure, it is referred to Appendix E.

5.1.2.3 Prediction of UNIFAC Group-Interaction Parameters

UNIFAC-MCM only contains parameters for the "pure" main groups, namely $\boldsymbol{\theta}_m$, $\boldsymbol{\beta}_m$, $\boldsymbol{\theta}_n$, $\boldsymbol{\beta}_n$, U_{mm}, and U_{nn}, which were fitted to the "group-mixture" data, namely the pseudo-data for $\ln \gamma_{mn}$, during the training of the model as described above. Based on the learned parameters, the group-interaction energies U_{mn} of all combinations of the considered main groups can be calculated based on Eq. (41), from which, in turn, the commonly used group-interaction parameters of UNIFAC a_{mn} and a_{nm} can be predicted from Eq. (40). Hence, a *complete* parameterization of UNIFAC regarding the first 50 main groups is obtained by this procedure, which can be used for predicting temperature- and concentration-dependent activity coefficients $\ln \gamma_i$ of all components i in any (binary or multi-component) mixture, if all components that make up the mixture can be segmented using the first 50 main groups of UNIFAC. The predicted complete set of a_{mn} (and of U_{mn}) are reported as a .csv file in Ref. [84]. Note that this set of a_{mn} is consistent in terms of fulfilling Eq. (40) as demanded by the lattice theory, which is in contrast to the previously available UNIFAC parameter tables that were obtained by fitting a_{mn} individually.

The latter also explains why a direct matrix factorization of the reported a_{mn} is not expedient, and instead the pseudo-data for $\ln \gamma_{mn}$ were used for training UNIFAC-MCM; the reported a_{mn} matrix simply lacks structure that could be exploited by the MCM.

5.1.3 Results and Discussion

In the following, the quality of UNIFAC-MCM is evaluated by considering predictions of vapor-liquid equilibria (VLE), which is probably the most important field in which activity coefficients are applied. As basis for this evaluation, all VLE data sets for binary mixtures from the Dortmund Data Bank (DDB) [135–137] that comply with the following conditions have been used:

- both components of the mixture can be built from the first 50 main groups of UNIFAC [12];
- the data set contains information on temperature, pressure, and composition of the liquid and vapor phase;
- the data set is labeled as "thermodynamically consistent" in the DDB, i.e., it fulfills area and point-to-point tests [138–140];
- Antoine parameters for calculating the pure-component vapor pressure at the temperature of the VLE are available in the DDB for both components;
- the pressure is not higher than 10 bar to justify the assumption of an ideal gas phase.

In the present version of the DDB, such VLE data are available for 2,246 distinct binary systems. This complete set of binary systems will be called "complete horizon" in the following.

The VLE were predicted using extended Raoult's law assuming an ideal vapor phase and a pressure independence of the chemical potentials in the liquid phase:

$$p_i^{\mathrm{s}}(T)\ x_i\ \gamma_i(T, x_i) = p\ y_i \tag{43}$$

For the calculations, the mole fractions x_i in the liquid phase as well as either the pressure p (for isobaric data sets) or the temperature T (for isothermal data sets) were specified, the pure component vapor pressure p_i^{s} was calculated with the Antoine equation using the parameters from the DDB, and the activity coefficients γ_i of the components in the liquid phase were predicted with UNIFAC-MCM. The mole fractions y_i in the vapor phase and the pressure p (for isothermal data sets) or the temperature T (for isobaric data sets) were then calculated from the system of equations resulting from applying Eq. (43) to both components. The results were compared to the experimental data from the DDB, with a focus on the gas phase mole fractions of the low-boiling component.

For comparison, the same calculations were also carried out with UNIFAC [12]; albeit, this is only possible for a subset of 2,068 systems from the complete horizon ("UNIFAC

horizon"). At a first glance, it may look disappointing that by using UNIFAC-MCM, with its substantially enlarged parameter table, only 178 additional systems for which data are available can be modeled. However, this is as expected: the lack of data on these systems has hindered the extension of the UNIFAC parameter table so far. Furthermore, also the commercial version UNIFAC-TUC has been used for comparison, which enabled predictions of VLE for 2,237 of the studied systems ("UNIFAC-TUC horizon"). The results from UNIFAC-TUC have been included in the comparison (even though it is not publicly available) for two reasons: firstly, it is the best available benchmark method and, secondly, it allows to evaluate the predictive performance UNIFAC-MCM also on systems that can not be modeled by UNIFAC, which is the basis of UNIFAC-MCM.

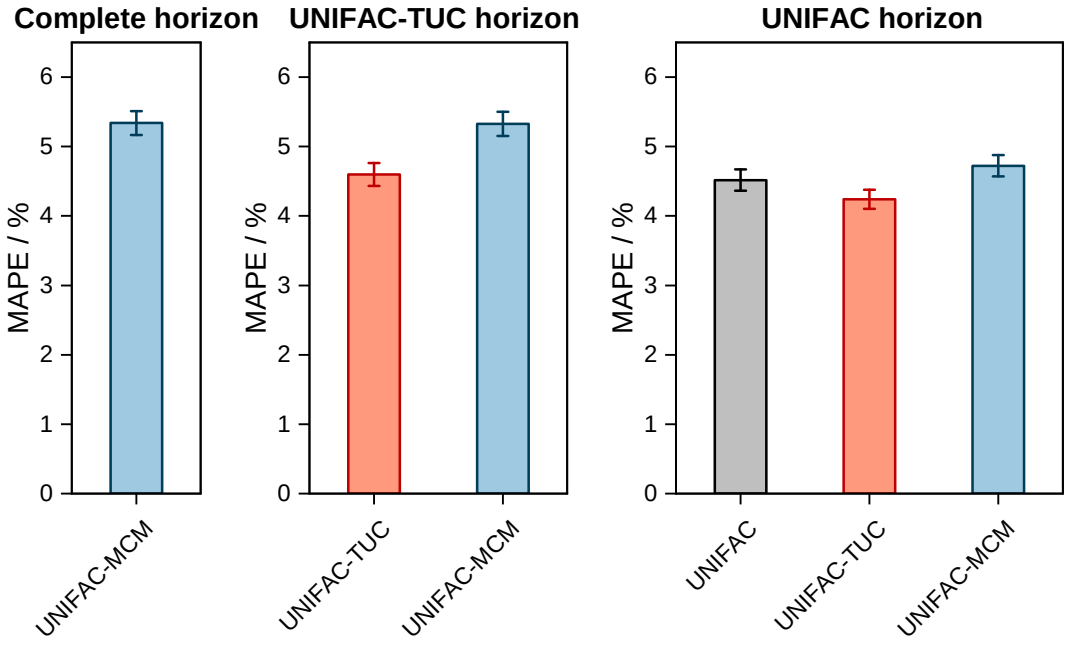


Figure 33: Mean Absolute Percentage Error (MAPE) of the predicted vapor-phase mole fractions of the low-boiling component in VLE with UNIFAC-MCM for the "complete horizon" (2,246 systems, left) and comparison to the commercial UNIFAC-TUC for the "UNIFAC-TUC horizon" (2,237 systems, middle), and to the public UNIFAC [12] for the "UNIFAC horizon" (2,068 systems, right). Error bars denote standard errors of the means.

The results are shown in Fig. 33, where the horizons in the three panels differ: in the left panel, it is the complete horizon, in the middle panel, it is the UNIFAC-TUC horizon, and in the right one, it is the smallest horizon, that of UNIFAC [12].

The results obtained with UNIFAC-MCM on the complete horizon are shown in Fig. 33 (left), where the Mean Absolute Percentage Error (MAPE) in y_i of the low-boiling component of the predictions with UNIFAC-MCM averaged over all 2,246 systems is plotted, which was calculated by comparing the UNIFAC-MCM predictions system-wise to the respective experimental data from the DDB. As the results indicate, UNIFAC-

MCM predicts the vapor-phase mole fractions for all 2,246 studied binary systems with an average error of 5.3%, which is not much larger than the typical uncertainty of experimental data for vapor-phase mole fractions. The MAPE of UNIFAC-MCM in the pressure p, averaged over all isothermal data sets from the complete horizon, is 5.0±0.2%; the MAPE in the absolute temperature T in K, averaged over all isobaric data sets from the complete horizon, is 0.48±0.02%.

In the middle panel of Fig. 33, the performance of MCM-UNIFAC is compared to that of UNIFAC-TUC, and in the right panel, it is compared to UNIFAC [12] as well as to UNIFAC-TUC. The highest accuracy among the three models is found for the commercial UNIFAC-TUC (MAPE of 4.6% on the UNIFAC-TUC horizon, cf. middle panel, and 4.2% on the UNIFAC horizon, cf. right panel), which is not surprising since a lot of effort has been put into refining its parameterization during the last decades. However, the scores of UNIFAC-MCM (MAPE of 5.3% on the UNIFAC-TUC horizon, cf. middle panel, and 4.7% on the UNIFAC horizon, cf. right panel) are only slightly worse than that of UNIFAC-TUC.

On the UNIFAC horizon, cf. Fig. 33 (right), the scores of UNIFAC-MCM (MAPE of 4.7%) and of the public UNIFAC (MAPE of 4.5%) are very similar. This demonstrates two things: first, that the additional flexibility of the UNIFAC model achieved by the inconsistent *individual* fitting of group-interaction parameters a_{mn} and a_{nm} compared to the sole physical consideration of group-interaction energies U_{mn} (including the like group-interaction energies U_{mm} and U_{nn}) is unnecessary; for the complete matrix of the considered 50 main groups of UNIFAC, there are 2,450 distinct group-interaction parameters a_{mn} and a_{nm}, but only 1,275 distinct group-interaction energies U_{mn} (including 50 like energies U_{mm}). And second, the MCM, which is at the heart of UNIFAC-MCM, is able to capture the structure within the unlike group-interaction energies using six latent parameters for each main group.

It is interesting to also study the performance of UNIFAC-MCM and UNIFAC-TUC only for those systems that *cannot* be modeled with UNIFAC [12]; this gives an impression of the performance of UNIFAC-MCM when applied for true predictions, namely for systems containing combinations of main groups for which no interaction parameters of UNIFAC are available, as it is unlikely that data on any of these systems were used in the development of UNIFAC [12], on which UNIFAC-MCM is based. In contrast, it may be assumed that basically all these additional VLE data were used for the development of UNIFAC-TUC, so that for UNIFAC-TUC, such a comparison shows basically only if the correlation of these additional data was successful. The respective results are presented in Fig. 34. Most of the systems within the complete horizon can be modeled not only with UNIFAC-MCM but also with UNIFAC-TUC. The few systems for which this is not the case, are treated separately in Fig. 34 (left panel).

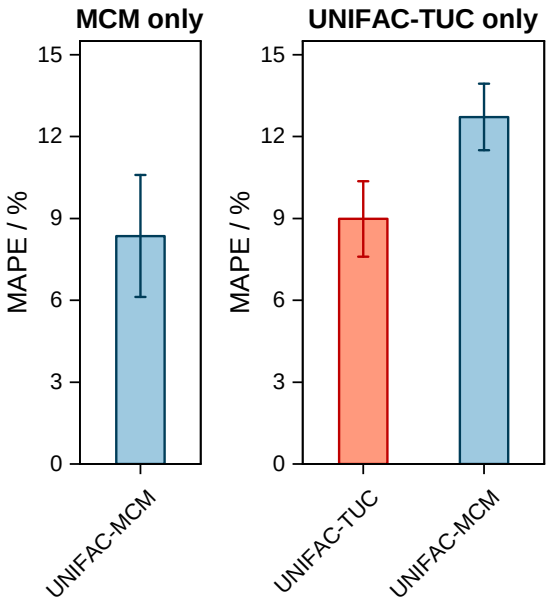


Figure 34: Mean Absolute Percentage Error (MAPE) of the predicted mole fraction of the low-boiling component in the vapor phase in VLE with UNIFAC-MCM for the systems that can only be modeled by UNIFAC-MCM (left, "MCM only", 9 systems), and those systems that can also be predicted with UNIFAC-TUC but not with UNIFAC (right, "UNIFAC-TUC only", 169 systems). Error bars denote standard errors of the means.

The first message from Fig. 34 is that the deviations increase compared to the ones shown in Fig. 33, which holds both for UNIFAC-TUC and UNIFAC-MCM. Averaged over all systems that can be modeled by both models (but not by UNIFAC), cf. Fig. 34 (right), the MAPE for UNIFAC-TUC is now 9.0%, that for UNIFAC-MCM is 12.7%. However, considering that the results from Fig. 34 obtained with UNIFAC-MCM are bold predictions, while those from UNIFAC-TUC are basically only correlations, the difference between both methods is unexpectedly small.

Comparing the results from Fig. 34 with those from Fig. 33 is most informative when referring to Fig. 33 (right), where the UNIFAC horizon is shown, because it then gives an impression on the changes when carrying out the comparison for complementary data sets: the UNIFAC horizon, for which the results are shown in Fig. 33 (right), covers all systems that can also be modeled by the public UNIFAC; Fig. 34, on the other hand, shows the results for all remaining systems from the data set, i.e., for the ones that *cannot* be modeled by the public UNIFAC.

Carrying out this comparison for UNIFAC-TUC (for which the results are correlations in both cases) clearly shows that the systems studied in Fig. 34 are more difficult to describe than those studied in Fig. 33 (right). The details of these additional difficulties are not discussed here, which can be related to different factors, including spotty and uncertain data (cf. also Fig. E.3 in Appendix E) as well as to the fact that many of

the respective systems contain components with special properties (highly halogenated or reactive components), which substantially complicates the accurate modeling with UNIFAC.

Hence, the results for UNIFAC-TUC indicate that most of the increase of the MAPE scores observed also for UNIFAC-MCM when going from Fig. 33 (right) to Fig. 34 is simply due to the increased difficulties in describing the data considered in Fig. 34, and, thus, cannot be attributed to a lack of predictive power.

It is noted here that the scope of the developed UNIFAC-MCM is much larger than can be demonstrated here, simply due to the fact that for many of the group-interaction parameters that can now be predicted, no experimental data for testing are available, cf. Fig. E.3 in Appendix E. An alternative representation of the results of UNIFAC-MCM in the form of histograms is given in Fig. E.4 in Appendix E.

Fig. 35 shows some typical examples for the prediction of vapor-liquid phase diagrams with UNIFAC-MCM and compares the results to those obtaiend with UNIFAC-TUC. Only systems that *cannot* be modeled by the public UNIFAC version were therefore chosen, such that the results of UNIFAC-MCM are true predictions. This is, again, not the case for UNIFAC-TUC, as the data shown in Fig. 35 were available for the development of the method. In all cases, UNIFAC-MCM represents the different types of phase behavior well.

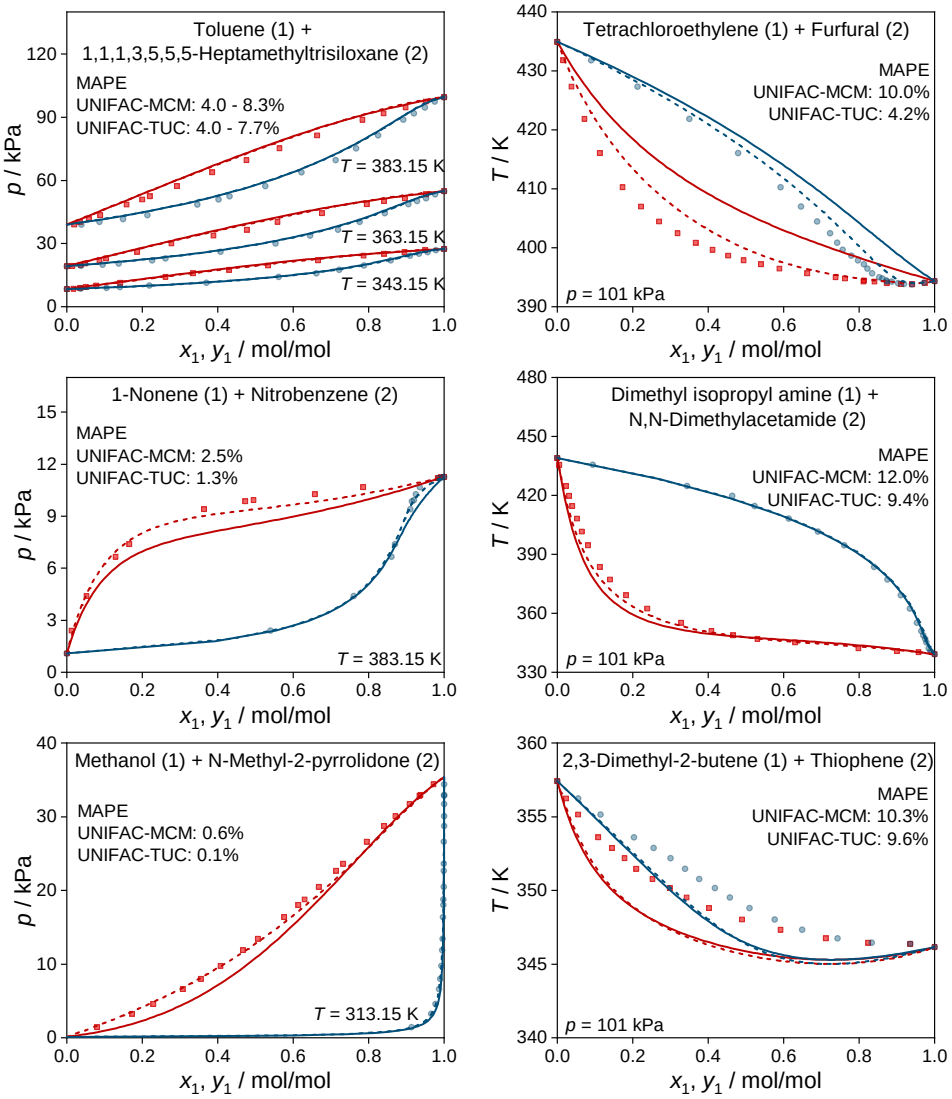


Figure 35: Prediction of vapor-liquid phase diagrams for binary systems with UNIFAC-MCM (solid lines) and UNIFAC-TUC (dashed lines) and comparison to experimental data from the DDB (symbols). For each system, the MAPE in the predicted vapor-phase mole fraction of the low-boiling component is given for both models. All shown systems can not be predicted with the public UNIFAC version. Blue: dew point curves. Red: bubble point curves.

Fig. 36 shows two further examples for the prediction of VLE phase diagrams with UNIFAC-MCM. The chosen systems can neither be modeled by the public UNIFAC, nor with the commercial UNIFAC-TUC due to missing group-interaction parameters in both models. An almost perfect agreement of the predictions with UNIFAC-MCM and the experimental data is observed, but it is also noted that systems with poorer agreement are found (cf. Fig. E.4 in Appendix E).

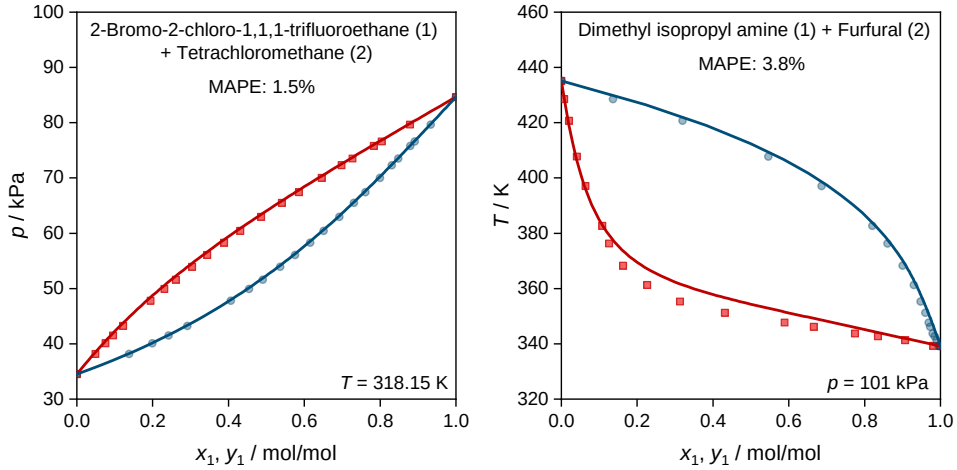


Figure 36: Prediction of vapor-liquid phase diagrams for binary systems with UNIFAC-MCM (lines) and comparison to experimental data from the DDB (symbols). For both system, the MAPE in the predicted vapor-phase mole fraction of the low-boiling component is given. Both systems can neither be predicted with the public UNIFAC version, nor with the commercial UNIFAC-TUC. Blue: dew point curves. Red: bubble point curves.

UNIFAC-MCM should in general be used in cases in which required group-interaction parameters of UNIFAC are missing, while in cases in which all parameters are available, their use is recommended. The reason is that UNIFAC-MCM is basically a derivate of UNIFAC, i.e., based on the available parameter tables, and it would only be by chance that it would be better than its basis for certain systems. However, it is emphasized that the differences between UNIFAC and UNIFAC-MCM are not expected to be large, as shown in Fig. 33.

5.1.4 Conclusions

Group-contribution methods for the prediction of thermophysical properties are highly important in chemical engineering. One of the most successful of these methods is UNIFAC. However, the applicability of UNIFAC is still substantially hampered by missing group-interaction parameters, which is in particular due to the lack of suitable mixture data for fitting the parameters. As a consequence, there are still significant gaps in the matrix in which these UNIFAC parameters are usually represented.

In the present chapter, an approach to *complete* the group-interaction parameter set of UNIFAC using a matrix completion method (MCM) from machine learning is presented. This approach, termed UNIFAC-MCM, was trained in a purely data-based manner solely on pseudo-data generated with UNIFAC and approximately doubles the number of available group-interaction parameters.

The performance of UNIFAC-MCM for the prediction of vapor-liquid equilibria (VLE) of 2,246 binary systems from the Dortmund Data Bank has been evaluated. This set can be divided into data that can be predicted with the public UNIFAC (2,068 systems) and data for which this is not the case, but which can be predicted with the developed UNIFAC-MCM (169 systems). The latter set is comparatively small, as the missing groups in UNIFAC are rather uncommon ones, i.e., only present in components for which only few data have been measured.

Where a direct comparison is possible, UNIFAC and UNIFAC-MCM show a similar performance. This alone is astonishing since UNIFAC-MCM is based only on *consistent group-interaction* energies, whereas in UNIFAC the number of the parameters to describe the pairwise interactions has almost been doubled, simply to increase the flexibility, which is, however, not well founded in the physical lattice theory from which UNIFAC was derived. For the systems for which UNIFAC can not be applied, the performance of UNIFAC-MCM is poorer but still acceptable, especially given the fact the this set contains basically only demanding systems, as also the commercial version UNIFAC-TUC, which was used for comparison here, shows significantly larger error scores.

This chapter has shown that working with consistent group-interaction energies is not only a feasible alternative to the common procedure of fitting UNIFAC parameters, but also a highly attractive one: a similar quality is obtained by a significantly smaller (approx. 50%) number of parameters, which promises a higher predictive performance and could be useful also for the fitting of new UNIFAC parameters in the future.

5.2 End-to-End Training on Thermodynamic Properties

5.2.1 UNIFAC 2.0

5.2.1.1 Introduction

Understanding the thermodynamic properties of mixtures is indispensable in chemical engineering and various related disciplines. However, the vast combinatorial diversity of mixtures makes it impossible to study each relevant mixture experimentally, necessitating reliable prediction methods. Group-contribution (GC) methods address this challenge by deconstructing components into structural groups, significantly reducing the number of parameters since the number of structural groups is much smaller than those of individual components. These methods rely on modeling pair interactions between these structural groups to describe mixture behavior. The effectiveness of GC methods hinges on selecting suitable groups and accurately determining their interaction parameters, both of which depend crucially on the database used for method development and parameterization.

Among GC methods, UNIFAC stands out as the most sophisticated and widely adopted approach for predicting activity coefficients in liquid mixtures. Since its introduction in 1975 [41], UNIFAC has undergone continuous refinement and improvement [12, 119–123], becoming integral to industrial process simulations. Available in both public [12] and commercial [141] formats, UNIFAC supports diverse applications, including variants like UNIFAC LLE [124] for predicting liquid-liquid equilibria. All UNIFAC variants rely on the same equations but differ in the number and type of groups considered and their parameterization. The process of finding suitable UNIFAC parameters was, in the past, sequential and based on a stepwise extension whenever data became available. This tedious process makes it very difficult to modify decisions taken at early steps.

This chapter addresses the challenges of updating and improving UNIFAC by leveraging modern computational techniques, aiming to enhance prediction accuracy and expand its applicability across a broader range of components and mixtures.

Throughout this chapter, the latest published version of UNIFAC is referenced. It was trained on a broad data basis focusing on vapor-liquid equilibrium data to develop a widely applicable model, not one for some specific purpose [12]. It is astonishing that, despite the importance of UNIFAC, this version is about 20 years old. The leading developers of UNIFAC have updated the method since then, but they have not disclosed these updates – they are only available for members of the UNIFAC-Consortium. One

might ask why no one else has updated this important method since then. The answer to this question is undoubtedly related to the considerable effort required to do this when the conventional strategy is used. Another issue is the accessibility of suitable data. For simplicity, the reference version of UNIFAC [12] will be labeled as UNIFAC 1.0 here.

UNIFAC describes the molar excess Gibbs energy, g^{E}, of a mixture as a function of temperature, T, and composition. From g^{E}, the activity coefficients of the components i, γ_i, in the mixture are obtained. UNIFAC contains group-specific parameters, namely, a size parameter (R_k) and a surface parameter (Q_k), as well as binary pair-interaction parameters (there are two for each group combination $a_{mn} \neq a_{nm}$, which will be often referred to simply as a_{mn}). UNIFAC 1.0 considers 54 *main groups*, subdivided into 113 *subgroups* [12].

Applying UNIFAC 1.0 to a given mixture requires the following: i) all components of the mixture must be decomposable into the 113 subgroups, ii) the parameters R_k and Q_k must be available for each relevant subgroup k, and iii) the pair-interaction parameters a_{mn} must be available for each binary combination of the relevant main groups m and n (all subgroups of a given main group share the same interaction parameters). The group parameters R_k and Q_k are available for all 113 groups [38], but interaction parameters a_{mn} are missing for many pairs of groups. Specifically, numbers for the interaction parameters are only available for 44% of all pairs of groups; Fig. F.1 in Appendix F illustrates this. The missing pair-interaction parameters, in some cases due to the challenging fitting process and in other cases due to the lack of experimental data for direct fitting, severely hampers the applicability of UNIFAC 1.0 (a single missing relevant parameter prevents the application of the model).

In this chapter, a new way of determining the interaction parameters of GC methods based on machine learning is introduced. The pair-interaction parameters can be treated as elements of a square matrix with dimensions 54×54, where the size corresponds to the number of structural groups. Since experimental data are only available for a fraction of the pair-interaction parameters, many entries of this matrix cannot be fitted directly, resulting in a matrix completion problem that can, in general, be solved by matrix completion methods (MCMs) [4, 14, 61]. As numbers for all entries are found, the problem of missing parameters does not exist anymore. In the MCM, so-called group features are determined for all groups from a fit to experimental data on activity coefficients. The entire data set is considered during the fit, and a well-defined learning algorithm is applied. This method replaces the sequential, intuitively guided procedure previously used to determine pair-interaction parameters. As the number of features to be determined scales linearly with the number of main groups N_{MG} ($\mathcal{O}(N_{\mathrm{MG}})$), it is much lower than the number of interaction parameters ($\mathcal{O}(N_{\mathrm{MG}}^2)$). Consequently,

the parameterization of the MCM is significantly more robust than a direct fit of the interaction parameters to the experimental data.

From the features of any two groups m and n of interest, the entries of the interaction parameter matrix a_{mn} are found by a simple matrix multiplication, resulting in a complete set of interaction parameters, thus facilitating the prediction of the activity coefficients for any binary mixture given its structural group composition at any temperature and concentration.

The result is UNIFAC 2.0, a hybrid model consisting of the framework of the physical UNIFAC model, in which an MCM from machine learning is embedded. While the MCM used for predicting missing interaction parameters from group-specific features is rather simple, UNIFAC 2.0 fully retains the non-linear UNIFAC equations, allowing it to also describe complex interactions between structural groups.

MCMs have already been employed for directly predicting thermodynamic properties of binary mixtures [8–10, 82, 83]. It has also been shown that MCMs are suitable for predicting pair-interaction parameters of components [84] and structural groups using synthetic training data (cf. Chapter 5.1). The synthetic training data in Chapter 5.1 were derived from the existing parameter tables of UNIFAC 1.0, providing a practical starting point. However, the limited prediction accuracy of this approach underscores the need for a more comprehensive approach. In this chapter, the first application of MCMs to the development of GC methods for predicting activity coefficients with direct end-to-end training on several hundred thousand experimental data points is presented.

5.2.1.2 Development of UNIFAC 2.0

5.2.1.2.1 General Framework

Fig. 37 illustrates UNIFAC 2.0 with end-to-end training of MCM features, which is compared to UNIFAC 1.0 with sequential parameter fitting. Both UNIFAC variants are based on the same structural groups and physical model equations. UNIFAC 2.0 was trained on experimental logarithmic activity coefficients ($\ln \gamma_i$) in binary mixtures derived from vapor-liquid equilibrium data for binary mixtures, cf. Section "Data" for details.

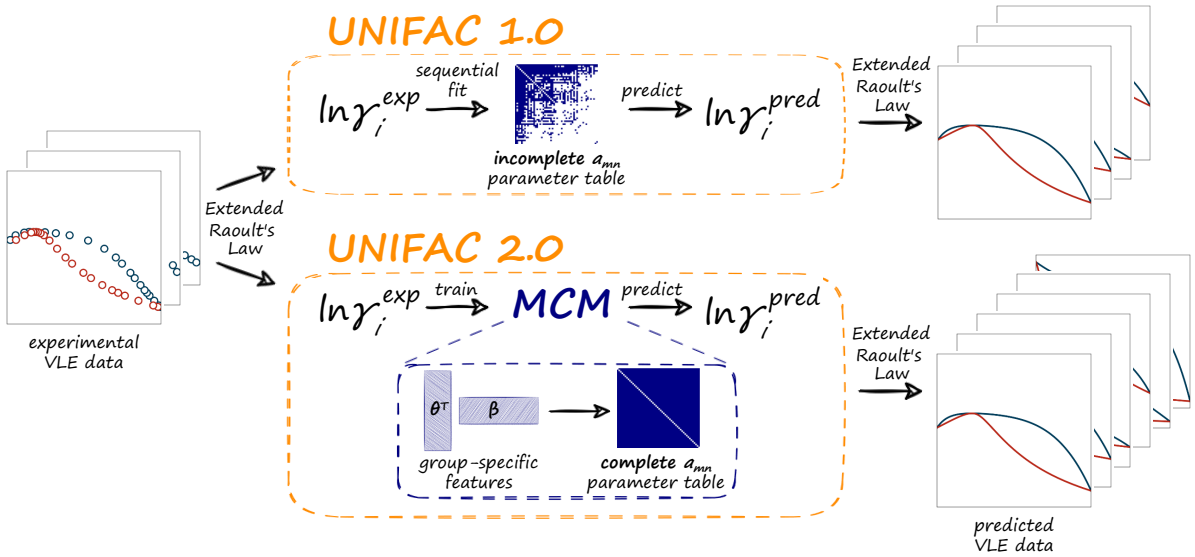


Figure 37: Comparison of UNIFAC 1.0 and UNIFAC 2.0. UNIFAC 1.0 relies on sequential parameter fitting guided by intuition, whereas UNIFAC 2.0 integrates a matrix completion method (MCM) for predicting pair-interaction parameters into the UNIFAC framework. UNIFAC 2.0 is trained end-to-end on experimental logarithmic activity coefficients ($\ln \gamma_i$) derived from binary vapor-liquid equilibrium (VLE) data. After training, the completed pair-interaction parameter matrix facilitates accurate predictions of phase diagrams for a wide range of binary or multi-component mixtures.

The MCM can only work if the available entries of the matrix are correlated. The MCM learns these correlations and represents them by the features. This enables the prediction of missing matrix entries through learned features. Each pair-interaction parameter a_{mn} is thereby modeled as follows:

$$a_{mn} = \boldsymbol{\theta}_m \cdot \boldsymbol{\beta}_n \tag{44}$$

Here, $\boldsymbol{\theta}_m$ and $\boldsymbol{\beta}_n$ are vectors of length K, with K representing the latent dimension, a hyperparameter that was determined in preliminary studies and set to $K = 8$. The feature vectors $\boldsymbol{\theta}_m$ and $\boldsymbol{\beta}_n$ are an abstract characterization of the structural groups determining their interactions with other groups.

A Bayesian approach is applied to train the model, treating each logarithmic activity coefficient $\ln \gamma_i$, each feature, and each interaction parameter a_{mn} as a random variable following a probability distribution, detailed further in the Section "Probabilistic Model". From the model training, a probability density is obtained for each a_{mn}, the mean of which is used to obtain the scalar value for each parameter. These scalar values are then used in all subsequent evaluations. The completed set of interaction parameters a_{mn}, derived from training on all considered binary data, and the subgroup-specific size

parameters R_k and Q_k for using UNIFAC 2.0 are provided freely in Ref. [54]. The size parameters are identical to those of the published UNIFAC 1.0 version.

The relevance of UNIFAC 2.0 becomes apparent when analyzing the applicability of UNIFAC 1.0 and 2.0 considering an example: the Dortmund data bank (DDB), which is the most extensive database for thermodynamic properties, presently lists 39,587 unique components that can be broken down into the published UNIFAC subgroups, which translates into more than 783 million possible binary mixtures. Of these binary mixtures, UNIFAC 1.0 is limited to predicting only 58% due to missing pair-interaction parameters, whereas UNIFAC 2.0 can be applied to all these mixtures. For multi-component mixtures, the fraction of mixtures that can only be predicted with UNIFAC 2.0 increases dramatically with an increasing number of components, as a mixture drops out if only a single parameter (pair) is missing.

Besides the hybrid model described above, a variant that is based on symmetrical pair-interaction energies $U_{mn} = U_{nm}$ between main groups instead of the asymmetric parameters a_{mn} was developed and tested. The symmetric model has fewer parameters and performs almost as well as the asymmetric model. The asymmetric model is reported on here, as it is the standard way to use UNIFAC, and the results can be implemented and used in a very simple manner. Details on the symmetric model are given in Appendix F. For a short background discussion of the two variants applied to component-wise pair interactions, see Ref. [84].

5.2.1.2.2 Probabilistic Model

The proposed probabilistic model integrates observations ($\ln \gamma_i$) and the latent variables (LVs) that characterize UNIFAC main groups ($\boldsymbol{\theta}_m$, $\boldsymbol{\beta}_n$) within a Bayesian framework, cf. Chapter 2. All $\ln \gamma_i$ and LVs are modeled as independent random variables. A standard normal distribution, i.e., a normal distribution with the mean $\mu = 0$ and the standard deviation $\sigma = 1$, is used as prior for each LV. The likelihood of observing $\ln \gamma_i$, given the LVs, follows a Cauchy distribution centered around the predicted activity coefficients $\ln \gamma_i^{\text{UNIFAC 2.0}}$ with scale parameter λ:

$$p(\ln \gamma_i \,|\, \boldsymbol{\theta}_m, \boldsymbol{\beta}_n) = \text{Cauchy}(\ln \gamma_i^{\text{UNIFAC 2.0}},\, \lambda) \tag{45}$$

where $\ln \gamma_i^{\text{UNIFAC 2.0}}$ is determined via the standard UNIFAC equations [12] using the predicted interaction parameters a_{mn}:

$$\ln \gamma_i^{\text{UNIFAC 2.0}} = \text{UNIFAC}(a_{mn}, R_k, Q_k, \boldsymbol{x}, T) \tag{46}$$

Here, R_k and Q_k are the subgroup-specific size parameters, T is the temperature, and $\boldsymbol{x}$ corresponds to the composition (expressed as mole fractions) of the considered binary mixture. The use of a Cauchy distribution for the likelihood is motivated by its robustness to outliers in the experimental data. Unlike the normal distribution, the heavy-tailed nature of the Cauchy distribution reduces the influence of extreme values, ensuring that the training process remains stable even when the data set contains flawed data points.

Written in Pyro, a probabilistic programming language based on Python and PyTorch support [24], the probabilistic model adopts stochastic variational inference (VI) [15] for posterior approximation. This approach leverages the Adam optimizer [25], with a learning rate of 0.15. A normal distribution is employed as the variational distribution, with all LVs being treated independently. During training, this approach facilitates learning variational parameters, specifically the mean and standard deviation, for each LV. Based on preliminary studies that have shown robust behavior in terms of predictive performance, the hyperparameters $K = 8$ and $\lambda = 0.4$ were chosen.

Post-training, the LVs inferred from the posterior enable, via Eqs. (44) and (46), the prediction of $\ln \gamma_i$ for any binary or multi-component mixture, including unstudied ones, whose components can be decomposed in the 113 UNIFAC subgroups.

5.2.1.2.3 Data

Experimental data on vapor-liquid equilibria (VLE) and activity coefficients at infinite dilution in binary mixtures were taken from the largest database for thermodynamic properties, the DDB [38]. In the preprocessing phase, data points identified as poor quality by the DDB were excluded, and the focus was narrowed to binary mixtures whose components can be decomposed into UNIFAC subgroups. Furthermore, only VLE data points from which the activity coefficients γ_i of components i in the mixture could be calculated using the extended Raoult's law (cf. Eq.(43)), assuming an ideal gas and neglecting the pressure dependence of the chemical potential in the liquid phase, were used. The VLE data were limited to pressures up to 10 bar.

5.2.1.3 Results

5.2.1.3.1 Overall Performance of UNIFAC 2.0

To evaluate the performance of UNIFAC 2.0 and compare it to that of the original UNIFAC 1.0, the mean absolute error (MAE) and the mean squared error (MSE) in the logarithmic activity coefficients $\ln \gamma_i$ are employed, which are calculated mixture-wise

(from the scores for each binary mixture) to ensure that each mixture is weighted equally in the final score and frequently measured mixtures do not lead to a false impression of the model quality.

In the following, focus is placed on the predictions of UNIFAC 2.0 obtained after training the hybrid model on all available data points from the database. This way for assessing the model has been chosen since this is likely also the case for UNIFAC 1.0, as the people maintaining UNIFAC and the DDB are essentially the same (although the exact training set of UNIFAC 1.0 has not been disclosed). Therefore, the comparison is considered fair. Nevertheless, as described in the following sections, two additional extrapolation tests were carried out with UNIFAC 2.0 to dispel doubts about its predictive capacity.

The performance of UNIFAC 2.0 on all available experimental data is shown in Fig. 38 and compared to UNIFAC 1.0. Since UNIFAC 2.0 has a more extensive scope than UNIFAC 1.0, a distinction is made: all data points that can be predicted with both methods are labeled as the "UNIFAC 1.0 horizon", whereas all data points that can only be predicted with UNIFAC 2.0 are labeled as "UNIFAC 2.0 only".

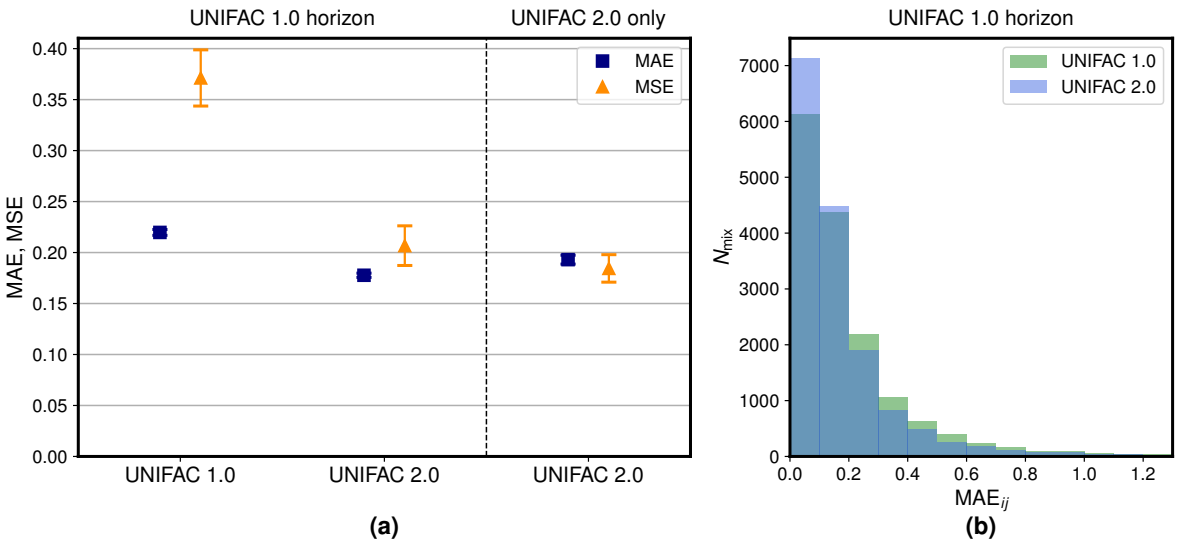


Figure 38: Comparison of results for $\ln \gamma_i$ with UNIFAC 1.0 and UNIFAC 2.0 for different data sets: the "UNIFAC 1.0 horizon" comprises 210,767 data points for 15,758 binary mixtures, while an additional 13,795 experimental data points for 2,957 binary mixtures can only be predicted with UNIFAC 2.0 ("UNIFAC 2.0 only"). (a) Mean absolute error (MAE) and mean squared error (MSE) of the model predictions. Error bars denote standard errors of the means. (b) Histogram of the number of binary mixtures N_{mix} that can be predicted with an MAE in a certain interval. The MAE range shown in panel (b) comprises 98.8% (UNIFAC 1.0) and 99.4% (UNIFAC 2.0) of all mixtures.

Fig. 38 (a) clearly shows the superior prediction accuracy of UNIFAC 2.0 over UNIFAC 1.0 in both error scores. The MSE can almost be halved compared to the original, demonstrating UNIFAC 2.0's effectiveness in reducing the occurrence of outliers. Table F.1 in Appendix F highlights the 20 binary mixtures with the largest improvement in prediction accuracy (MSE) achieved by UNIFAC 2.0 compared to UNIFAC 1.0. Notably, mixtures involving methoxy groups paired with silane groups and those with water paired with chlorinated aromatic components show significant improvements, indicating that these specific interactions benefit greatly from the updated parameters in UNIFAC 2.0. Even more importantly, the new method not only improves accuracy for data points within the predictive range of UNIFAC 1.0, but it also maintains this accuracy for data points beyond the scope of UNIFAC 1.0, cf. the results for the "UNIFAC 2.0 only" set.

In Fig. 38 (b), a detailed analysis of the MAE for the UNIFAC 1.0 horizon in the form of a histogram of individual binary mixture scores is shown. It underpins that UNIFAC 2.0 achieves an exceptional prediction accuracy: for 7,133 mixtures, the MAE is below 0.1, and thereby in the range of the experimental uncertainty. This accuracy is achieved for only 6,133 mixtures with UNIFAC 1.0.

The activity coefficients obtained by UNIFAC 2.0 can be used directly to predict phase equilibria of mixtures, which are at the core of many tasks in chemical engineering. For instance, vapor-liquid phase diagrams are crucial for designing and optimizing distillation processes, where the separation efficiency relies on accurate predictions of boiling and dew points. They also play a key role in azeotropic and extractive distillation, where deviations from ideality must be accurately modeled in order to select suitable entrainers. Beyond distillation, they are also directly applicable in absorption and stripping processes, where the vapor-liquid phase equilibrium determines the efficiency of gas capture or solvent recovery. Fig. 39 shows six examples of isothermal vapor-liquid phase diagrams predicted by UNIFAC 2.0, cf. Section "Data" for computational details. All six mixtures are part of the "UNIFAC 2.0 only" set, i.e., they cannot be modeled with the original UNIFAC 1.0. UNIFAC 2.0 accurately describes the phase behavior of all these mixtures. The examples shown in Fig. 39 represent typical cases and were selected to cover different types of phase behavior, ranging from small deviations of the ideal behavior to low-boiling azeotropes.

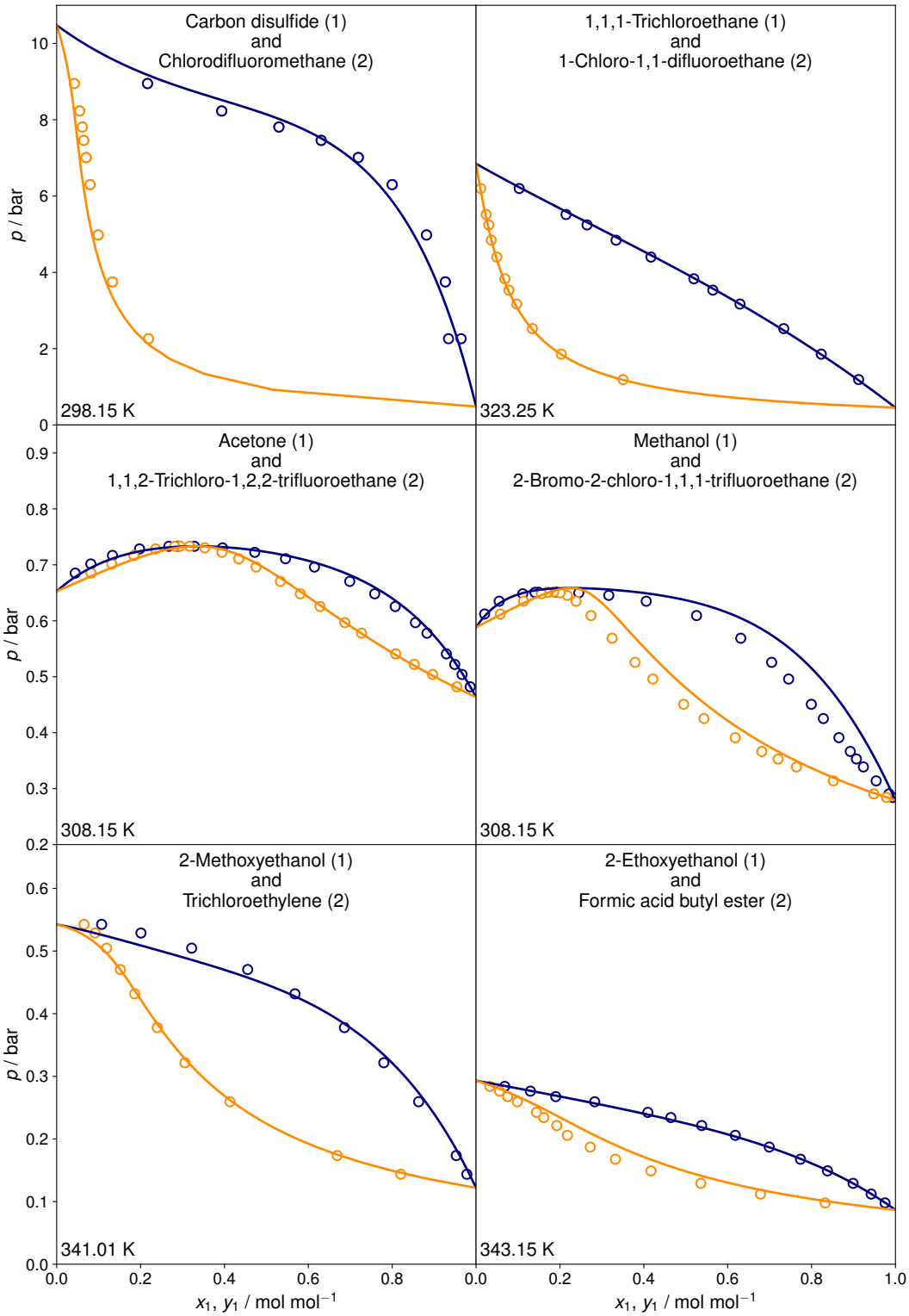


Figure 39: Prediction of isothermal vapor–liquid phase diagrams for binary mixtures with UNIFAC 2.0 (lines) and comparison to experimental data from the DDB (symbols). Blue: bubble point curves. Orange: dew point curves.

Furthermore, although no data on multi-component mixtures were used for training UNIFAC 2.0, the underlying physical framework of UNIFAC also enables predictions

for such mixtures. As examples, Fig. 40 shows isothermal vapor-liquid phase diagrams for two ternary mixtures selected from the "UNIFAC 2.0 only" set, i.e., for which UNIFAC 1.0 is not applicable. For each data point, the temperature and the liquid-phase composition (blue symbols in Fig. 40) were specified and used to predict the corresponding vapor-phase composition in equilibrium with UNIFAC 2.0 (shown as filled orange symbols), which was then compared to the experimentally determined vapor-phase composition (open orange symbols). Excellent accuracy is found.

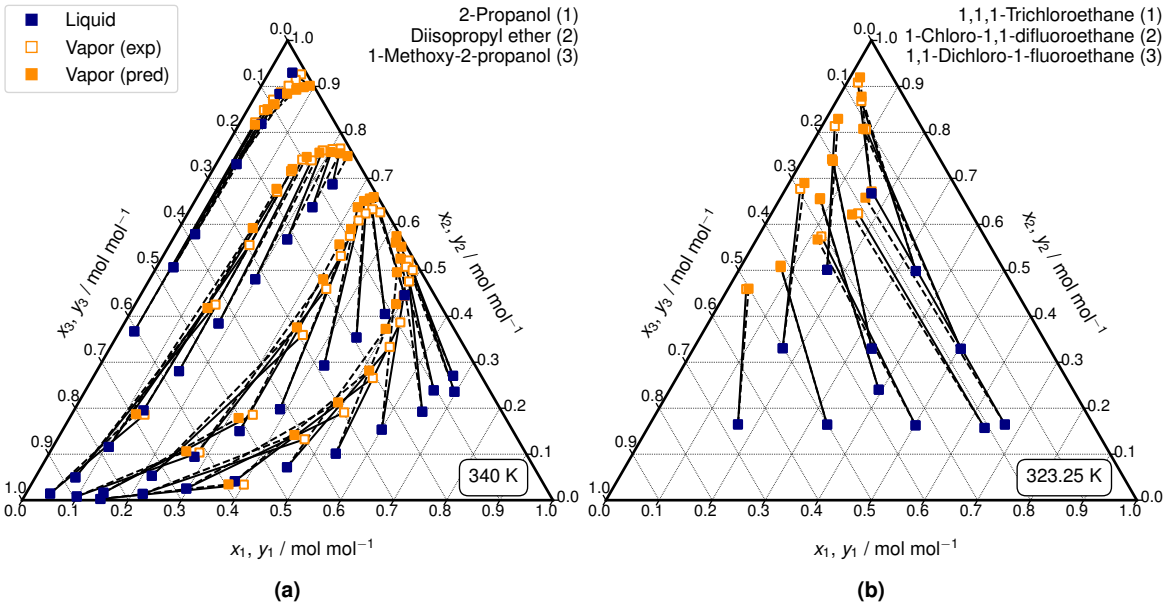


Figure 40: Prediction of isothermal vapor-liquid phase diagrams for ternary mixtures with UNIFAC 2.0 (pred) and comparison to experimental data (exp) from the DDB. The temperature and the composition of the liquid phase were specified, and the composition of the corresponding vapor phase in equilibrium was predicted. Solid lines are experimental conodes, dashed lines are predicted conodes.

The results demonstrate the very good performance of UNIFAC 2.0, which outperforms UNIFAC 1.0 not only in terms of applicability by closing all gaps in its parameter table but even in terms of prediction accuracy. This highlights UNIFAC 2.0 as a compelling approach to predicting activity coefficients, particularly as it retains the classic UNIFAC framework. This retention facilitates straightforward implementation in process simulators, ensuring broad accessibility and automatic applicability to multi-component mixtures – a significant advantage over other state-of-the-art machine learning approaches. Among these, HANNA, a recently developed hard-constraint neural network [53], is, to the author's knowledge, currently the most accurate model for predicting activity coefficients in binary mixtures. HANNA's accuracy is achieved through a much more flexible architecture, using more than 70 times the number of parameters compared to UNIFAC 2.0, complicating its direct use in process simulators. Furthermore, HANNA

is presently restricted to binary mixtures, whereas UNIFAC 2.0 can intrinsically and consistently extrapolate to multi-component mixtures. These trade-offs highlight the complementary strengths of UNIFAC 2.0 and other machine learning approaches like HANNA, which address different aspects of activity coefficient prediction and meet different user needs.

5.2.1.3.2 Extrapolation to Unseen Components

In a study to evaluate the capacity of UNIFAC 2.0 to extrapolate to unseen components, 100 randomly selected components were intentionally excluded from the training by withholding all data points for systems containing any of these components from the training set and using the systems removed from the training set as the test set. This test set contains 27,287 data points and covers 2,603 different binary mixtures. The results for this test set are shown in Fig. 41, which, again, contains the result from UNIFAC 1.0 for comparison.

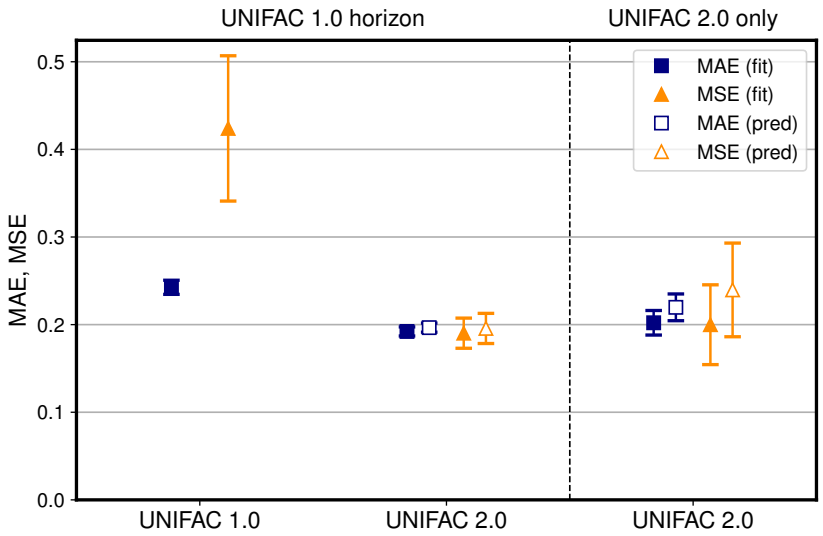


Figure 41: Mean absolute error (MAE) and mean squared error (MSE) of the predicted $\ln \gamma_i$ of mixtures containing unseen components with UNIFAC 2.0 (pred). For comparison, the results of UNIFAC 2.0 trained on all experimental data and UNIFAC 1.0 are also shown (fit). The "UNIFAC 1.0 horizon" comprises 25,998 data points for 2,202 binary mixtures, while an additional 1,289 experimental data points for 401 binary mixtures can only be predicted by UNIFAC 2.0 ("UNIFAC 2.0 only"). Error bars denote standard errors of the means.

Fig. 41 shows that the accuracy of the true predictions with UNIFAC 2.0 obtained by withholding the test data during the training (open symbols) is only marginally lower than that of the UNIFAC 2.0 version that was trained on all data points (closed

symbols); this holds for both the "UNIFAC 1.0 horizon" and the "UNIFAC 2.0 only" data sets. Furthermore, also in this true predictive test case, UNIFAC 2.0 outperforms UNIFAC 1.0, especially considering the MSE, even though it is likely that UNIFAC 1.0 has been trained on most of the test data points, as discussed above. These findings highlight, on the one hand, the robustness of UNIFAC 2.0 and, on the other hand, the predictive qualities of this hybrid model.

5.2.1.3.3 Extrapolation to Unseen Pair-Interaction Parameters

Another, even more challenging, test was carried out by randomly choosing 100 combinations of UNIFAC main groups for which experimental data are available and withholding the data on all systems in which any of the chosen combinations of groups occurs from the training of UNIFAC 2.0. In this way, the capacity of the hybrid model to predict pair-interaction parameters a_{mn} that cannot be obtained by direct fitting is investigated. For each of the 100 selected main group combinations, illustrated in Fig. F.4 in Appendix F, a test set was created that includes the data for those systems in which the selected group combination occurs. All other data points were used to train the model, and the predictions on the test set were evaluated. This process was repeated for all selected main group combinations. MAE and MSE were calculated for each test set. Fig. 42 shows the average error scores over all 100 test sets. Again, the results are compared to those of UNIFAC 1.0 and the UNIFAC 2.0 version trained on all data points. Note that the 100 test sets vary strongly in the number of data points and different binary mixtures, as shown in Table F.2 in Appendix F. This table also includes the MAE for each individual test set.

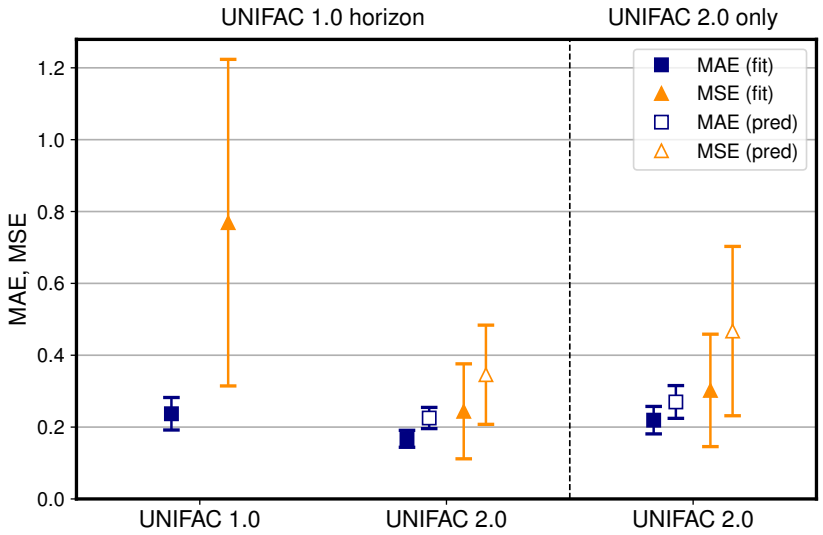


Figure 42: Mean absolute error (MAE) and mean squared error (MSE) of the predicted $\ln \gamma_i$ averaged over 100 test sets with UNIFAC 2.0 (pred). The test sets were created by selecting all data points for which a specific interaction parameter a_{mn} is relevant, cf. F.2 in Appendix F. The results for UNIFAC 2.0 trained on all experimental data and UNIFAC 1.0 are shown for comparison (fit). Error bars denote standard errors of the means.

The comparison of the UNIFAC 2.0 predictions to the UNIFAC 1.0 predictions on the "UNIFAC 1.0 horizon" in Fig. 42 reveals that the *truly predicted* pair-interaction parameters of UNIFAC 2.0 outperform those of UNIFAC 1.0, which were presumably *fitted* to the experimental data used for evaluation here; this is particularly evident considering the MSE. When comparing the true predictions with UNIFAC 2.0 (open symbols) to those of UNIFAC 2.0 trained on the whole experimental database (full symbols), a slight reduction in prediction accuracy is observed, as expected. However, the differences are small, which demonstrates the robustness of UNIFAC 2.0. The increased standard error associated with the MSE for UNIFAC 1.0 can be attributed to individual test sets for which the predictions are extremely poor.

The results of these tests demonstrate the capability of UNIFAC 2.0 to accurately predict pair-interaction parameters, which enormously increases the scope of this group-contribution method. UNIFAC 2.0 is not only much more applicable than UNIFAC 1.0, but its predictions are also more accurate, as shown by the comparison on the shared horizon. Hence, UNIFAC 2.0 should not only be used when UNIFAC 1.0 cannot be applied, but it should replace UNIFAC 1.0 as the default method for predicting activity coefficients. The fact that UNIFAC 2.0 performs better than UNIFAC 1.0 as measured by lumped criteria, such as the MAE and MSE, that have been used here for describing the performance on a broad database does not exclude, of course, that for specific systems, UNIFAC 1.0 may give better results. Implementing UNIFAC 2.0 is as simple as

possible: one must only substitute the original (incomplete) UNIFAC parameter table, e.g., in an established process simulator, with the completed one, which are provided in Ref. [54]. This ease of implementation clearly distinguishes UNIFAC 2.0 from other machine learning methods for property prediction.

5.2.1.4 Conclusions

Group-contribution (GC) methods are widely used workhorses for the prediction of thermodynamic properties of materials. Here, the focus is placed on how they can be combined with methods from machine learning to obtain hybrid models that outperform their physical parent models. This is demonstrated here for the GC model UNIFAC for predicting activity coefficients in liquid mixtures. UNIFAC is one of the most important GC methods, broadly used in engineering, and implemented in basically all process simulation packages. Like most GC methods for predicting properties of mixtures, UNIFAC is based on the concept of group pair interactions. It is demonstrated that these pair interactions can be learned and predicted with matrix completion methods (MCM) from machine learning. The resulting new hybrid model, UNIFAC 2.0, is systematically compared to its physical parent model, UNIFAC 1.0. In contrast to the UNIFAC 1.0 parameter table, which has significant gaps, the parameter table of UNIFAC 2.0 obtained from the MCM has no gaps, leading to a substantial increase in the range of applicability. One could expect to have to pay for this increase in applicability with a deterioration of the accuracy of the predictions - but this is not the case: UNIFAC 2.0 is better than its parent model in both regards.

The hybrid approach described here also has essential advantages regarding the workflow: as the physical framework is kept, the new model can be implemented very easily in existing software packages; only parameter tables have to be updated to use its advantages. The full UNIFAC 2.0 parameter table is provided in Ref. [54]. Furthermore, the end-to-end training of the hybrid model to experimental data can be carried out in an automated manner so that updates can be supplied easily if new data become available or targets shift; also, tailored versions of the model, adapted to special needs, can be obtained easily.

5.2.2 Modified UNIFAC 2.0

5.2.2.1 Introduction

Understanding the thermodynamic properties of mixtures is essential for chemical engineering. Due to the impracticality of studying each relevant mixture experimentally, reliable prediction methods are crucial. Group-contribution (GC) methods offer an efficient solution by decomposing molecules into structural groups, significantly reducing the number of parameters and enabling extrapolations to unstudied components and mixtures. The most successful GC method in chemical engineering is probably UNIFAC [41], which is available in different versions [12, 13, 124, 141]. UNIFAC is a model for predicting the excess Gibbs energy of mixtures and derived properties, such as activity coefficients and excess enthalpies. It has been widely adopted for describing reaction and phase equilibria in mixtures and is implemented in all relevant process simulators [142–144].

However, UNIFAC has important drawbacks: Firstly, the most comprehensive versions of UNIFAC, namely, original UNIFAC [12] and modified UNIFAC (Dortmund) [13], have been regularly updated, but only up to 2003 [12] and 2016 [13], respectively. Since then, the work on UNIFAC updates has continued, but only commercially within the so-called UNIFAC-Consortium (TUC) [141], so the latest UNIFAC versions are not publicly available. Furthermore, the applicability of all UNIFAC versions, including the commercial ones, is limited by the availability of pair-interaction parameters between structural groups. These parameters are derived from vapor-liquid equilibrium (VLE) and other thermodynamic data of mixtures, leaving substantial gaps when no suitable training data are available, severely hampering the applicability of UNIFAC.

Compared to the original UNIFAC [12], in which two parameters are used to describe the interactions between a given pair of groups, modified UNIFAC [13] considers the temperature dependence of these parameters by a simple function, leading to up to six parameters that can be adjusted for a given pair of groups. This increased flexibility often improves accuracy in describing different mixtures, making modified UNIFAC (Dortmund) arguably the best GC method presently available. For simplicity, the latest public version of modified UNIFAC (Dortmund), used here as the reference, will be labeled as mod. UNIFAC 1.0. Mod. UNIFAC 1.0 considers 63 *main groups*, subdivided into 125 *subgroups*. While each subgroup k has individual size parameters describing their surface area (Q_k) and volume (R_k), which are reported for all 125 defined subgroups, pair-interaction parameters are defined between main groups m and n. In the current parameterization of mod. UNIFAC 1.0, these interaction parameters are reported for only 39% of all possible pairs of main groups; Fig. G.1 in Appendix G illustrates this. This situation significantly hampers the applicability of mod. UNIFAC 1.0 since a single

missing group pair-interaction parameter for a given mixture prevents the use of the method.

Consequently, the pair-interaction parameters of mod. UNIFAC 1.0, which are asymmetric ($a_{mn} \neq a_{nm}$, $b_{mn} \neq b_{nm}$, $c_{mn} \neq c_{nm}$), can be arranged in (sparsely filled) matrices, making the prediction of the missing parameters a matrix completion problem, for which matrix completion methods (MCMs) from machine learning (ML) [4, 14] can be used. The applicability of MCMs in thermodynamics has been demonstrated in Refs. [8–10, 51, 82–84], where MCMs have been developed to predict different thermodynamic properties of mixtures [8–10, 82, 83] and different types of pair-interaction parameters [51, 84]. Most importantly, UNIFAC 2.0 has been introduced (cf. Chapter 5.2.1), a hybrid model that embeds an MCM into the framework of the original UNIFAC model [12]. Through this integration, the MCM predicts the missing pair-interaction parameters between the main groups of original UNIFAC. UNIFAC 2.0 was trained on experimental activity coefficients derived from binary VLE data and limiting activity coefficient data taken from the Dortmund Data Bank (DDB) [38] in an end-to-end manner, avoiding the sequential and often intuitive approaches that have characterized the traditional fitting process of UNIFAC. It has been demonstrated in Chapter 5.2.1 that the hybrid UNIFAC 2.0, based on a learned completed pair-interaction parameter table, outperforms the original UNIFAC method in terms of scope and accuracy.

In this chapter, the concept of embedding an MCM in GC methods is transferred from original UNIFAC to mod. UNIFAC and mod. UNIFAC 2.0 is introduced. Similar to UNIFAC 2.0, mod. UNIFAC 2.0 exhibits complete pair-interaction parameterizations and was trained end-to-end on an extensive database of more than 500,000 data points from the DDB. As the consideration of the temperature dependence of the group interactions makes mod. UNIFAC more flexible, experimental data on the excess enthalpy, in addition to data on activity coefficients, have been included in the training process of mod. UNIFAC 2.0.

By retaining the mod. UNIFAC equations, mod. UNIFAC 2.0 maintains the high accessibility of the original model and can easily be implemented in process simulators by simply replacing the parameter sets with the ones freely provided in Ref. [145]. At the same time, mod. UNIFAC 2.0 eliminates the most significant limitation of the original model by filling all gaps in the pair-interaction parameter tables, tremendously increasing the applicability to any mixture whose components can be represented by the presently defined structural groups. The subgroup-specific size parameters R_k and Q_k for using mod. UNIFAC 2.0, which are identical to those of the published mod. UNIFAC 1.0 version, are also provided in Ref. [145].

5.2.2.2 Development of Mod. UNIFAC 2.0

5.2.2.2.1 General Framework

Fig. 43 illustrates how mod. UNIFAC 2.0 was developed by embedding an MCM into the mod. UNIFAC framework. The resulting method was trained end-to-end on experimental logarithmic activity coefficients ($\ln \gamma_i$) and excess enthalpies (h^{E}) in binary mixtures. The $\ln \gamma_i$ were obtained from the limiting activity coefficient database of the DDB and derived from binary VLE data, cf. Section "Data" for details. Mod. UNIFAC 2.0 is compared here to mod. UNIFAC 1.0, which uses the same structural groups and physical model equations as mod. UNIFAC 2.0 but whose parameters were obtained by sequential parameter fitting on a data basis that includes only data taken before 2016. Additionally, mod. UNIFAC 1.0 was trained on additional mixture properties beyond those included here [13, 72].

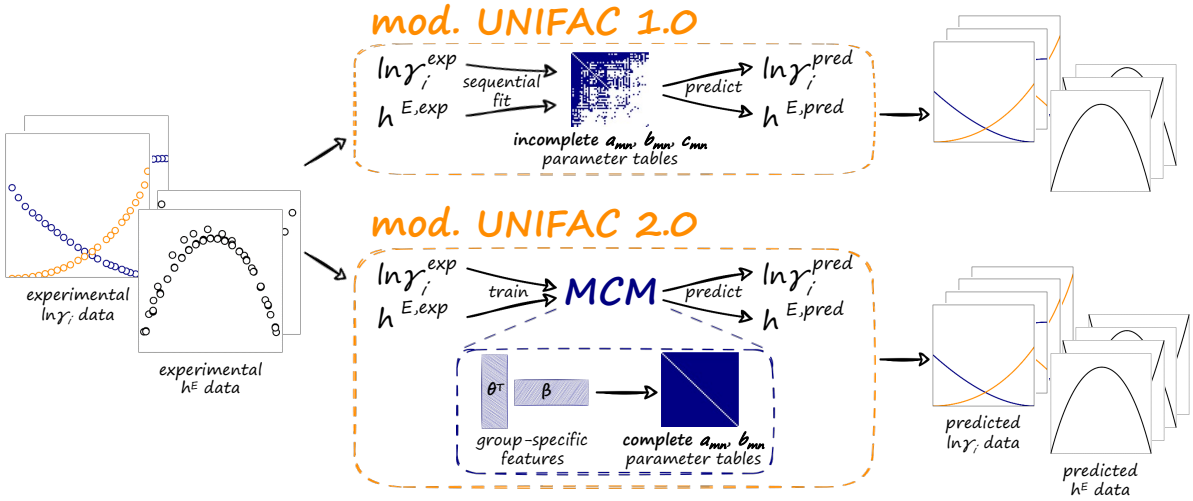


Figure 43: Comparison of mod. UNIFAC 1.0 [13] and mod. UNIFAC 2.0. Mod. UNIFAC 1.0 relies on sequential parameter fitting, whereas mod. UNIFAC 2.0 integrates a matrix completion method (MCM) for predicting pair-interaction parameters into the mod. UNIFAC framework. Mod. UNIFAC 2.0 was trained end-to-end on experimental logarithmic activity coefficients ($\ln \gamma_i$) and excess enthalpy (h^{E}) data. After training, the completed pair-interaction parameter matrices facilitate predictions of thermodynamic properties for a vast range of binary and multi-component mixtures.

Mod. UNIFAC 1.0 extends the parameter Ψ_{nm} of the original UNIFAC model by introducing a temperature dependence through the additional interaction parameters b_{mn} and c_{mn}:

$$\Psi_{nm} = \exp\left(-\frac{a_{nm} + b_{nm}T + c_{nm}T^2}{T}\right) \tag{47}$$

Setting $b_{mn} = c_{mn} = 0$ results in the original UNIFAC definition of Ψ_{nm} [12].

However, in mod. UNIFAC 1.0, c_{mn} parameters were fitted for only very few pairs of groups, and $c_{mn} = 0$ is used for most group combinations. Therefore, only a_{mn} and b_{mn} are used in mod. UNIFAC 2.0, which are modeled by an MCM trained to decompose the two matrices containing the parameters a_{mn} and b_{mn}, respectively, into the product of two respective feature matrices. Each pair-interaction parameter is thereby modeled as:

$$a_{mn} = \boldsymbol{\theta}_m^a \cdot \boldsymbol{\beta}_n^a \tag{48}$$

$$b_{mn} = \boldsymbol{\theta}_m^b \cdot \boldsymbol{\beta}_n^b \tag{49}$$

Here, $\boldsymbol{\theta}_m^a$, $\boldsymbol{\theta}_m^b$, $\boldsymbol{\beta}_n^a$, and $\boldsymbol{\beta}_n^b$ are vectors of length K, where K is called latent dimension. This hyperparameter was determined in preliminary studies and set to $K = 8$. For simplicity, the feature vectors are collectively referred to as θ and β in the following.

All parameters of mod. UNIFAC 2.0 are learned *simultaneously*, which is in sharp contrast to the sequential approach used in the original model. Mod. UNIFAC 2.0 was trained within a Bayesian framework, treating each experimental data point ($\ln \gamma_i$, h^{E}), feature (θ, β), and interaction parameter (a_{mn}, b_{mn}) as random variables drawn from probability distributions, cf. Chapter 2.

The prior for all features is a standard normal distribution, $\mathcal{N}(0, 1)$, which is uninformative and introduces no bias toward specific feature values, except for discouraging very large values, thereby serving as a kind of regularization. This choice provides a simple and effective starting point for learning features from the empirical data.

The likelihood defines the probability of observing the data ($\ln \gamma_i^{\mathrm{exp}}$ and $h^{\mathrm{E,exp}}$) given the features. It is modeled using a Cauchy distribution centered around the predicted values $\ln \gamma_i^{\mathrm{pred}}$ and $h^{\mathrm{E,pred}}$, respectively:

$$p(\ln \gamma_i^{\mathrm{exp}} | \theta, \beta) = \mathrm{Cauchy}(\ln \gamma_i^{\mathrm{pred}}, \lambda) \tag{50}$$

$$p(h^{\mathrm{E,exp}} | \theta, \beta) = \mathrm{Cauchy}(h^{\mathrm{E,pred}}, \lambda) \tag{51}$$

where λ is the scale parameter of the Cauchy distribution, which was set to $\lambda = 0.4$ as in Chapter 5.2.1. Predicted values for $\ln \gamma_i^{\mathrm{pred}}$ and $h^{\mathrm{E,pred}}$ are obtained using the standard mod. UNIFAC equations, which are fully described in Refs. [42, 146]:

$$\ln \gamma_i^{\mathrm{pred}} = \text{mod. UNIFAC}(a_{mn}, b_{mn}, R_k, Q_k, \boldsymbol{x}, T) \tag{52}$$

$$h^{\mathrm{E,pred}} = -RT^2 \sum_{i=1}^{N} x_i \left(\frac{\partial \ln \gamma_i^{\mathrm{pred}}}{\partial T} \right)_{p,\boldsymbol{x}} \tag{53}$$

where $\boldsymbol{x}$ is the composition vector (for binary mixtures, this reduces to x_1), T is the temperature, and a_{mn} and b_{mn} are the predicted pair-interaction parameters of mod. UNIFAC 2.0 calculated from the learned features according to Eqs. (48) and (49).

Using Pyro, a probabilistic programming language written in Python and supported by PyTorch [24], the posterior has been approximated using stochastic variational inference (VI) under the mean-field assumption [15], where all features are considered independent, and a normal variational distribution approximates each. During this step, the evidence lower bound (ELBO) was maximized using the Adam optimizer [25] with a learning rate of 0.15, ensuring efficient and scalable learning over the large experimental data set.

The result of training mod. UNIFAC 2.0 is a learned probability density for each feature, from which the means were used to calculate the final pair-interaction parameters (cf. Eqs. (48) and (49)), which are subsequently plugged into the mod. UNIFAC equations [42, 146] to give predictions for unstudied activity coefficients.

The complete final set of pair-interaction parameters – derived from training mod. UNIFAC 2.0 on the entire database (see Section "Data") – is freely available in Ref. [145] as .csv files. Additionally, the subgroup-specific size parameters R_k and Q_k are provided, which are identical to the published mod. UNIFAC 1.0 version [13].

5.2.2.2.2 Data

Experimental data for activity coefficients γ_i and excess enthalpies h^{E} in binary mixtures were used for training mod. UNIFAC 2.0. All data were taken from the most extensive database for thermodynamic properties, the DDB [142]. During preprocessing, data points that were considered to be of low quality by the DDB were excluded. Additionally, the selection was restricted to binary mixtures whose components could be decomposed into the mod. UNIFAC subgroups. Moreover, the VLE data were limited to pressures up to 10 bar.

After preprocessing, the h^{E} data set comprises 259,707 data points for 8,735 binary mixtures. The data set for γ_i consists of 243,257 data points for 21,452 binary mixtures, which was obtained by combining 68,642 data points for limiting activity coefficients and 174,615 data points calculated from VLE data using the extended Raoult's law assuming an ideal gas phase[2] and neglecting the pressure dependence of the chemical potential in the liquid phase, cf. Eq.(43).

[2] At elevated pressures of up to 10 bar, some deviations from this assumption have to be expected. The 10 bar limit was chosen as a compromise between limiting these deviations and losing interesting systems from the database. Fugacity coefficients to correct for the non-ideality of the gas phase have been neglected for computational reasons.

5.2.2.3 Results and Discussion

5.2.2.3.1 Overall Performance of Mod. UNIFAC 2.0

For evaluating the performance of mod. UNIFAC 2.0 in predicting activity coefficients $\ln \gamma_i$ and excess enthalpies h^{E}, the mean absolute error (MAE) for each binary mixture is used and the results are represented in box plots, as shown in Figs. 44 (for $\ln \gamma_i$) and 45 (for h^{E}). These plots also contain the corresponding results of mod. UNIFAC 1.0, evaluated on the same basis, for comparison. The results shown in these figures were obtained with a mod. UNIFAC 2.0 version trained on all available experimental data in the database. However, as detailed in the subsequent sections, two extrapolation tests have also been performed by withholding parts of the data during training to demonstrate and validate the predictive capacities of mod. UNIFAC 2.0.

Although the exact training set for mod. UNIFAC 1.0 has not been disclosed, it is reasonable to assume that the experimental data used in this chapter are similar to the data used for its parameterization, which supports a fair comparison in Figs. 44 and 45. Note that the comparison between mod. UNIFAC 1.0 and mod. UNIFAC 2.0 is carried out on the "mod. UNIFAC 1.0 horizon", i.e., only those mixtures from the data set that can be modeled with the incomplete parameter set of mod. UNIFAC 1.0. Since mod. UNIFAC 2.0, with its completed parameter set, has a much larger scope, its performance is additionally evaluated on those mixtures that cannot be predicted with mod. UNIFAC 1.0, labeled as the "mod. UNIFAC 2.0 only" data set in Figs. 44 and 45.

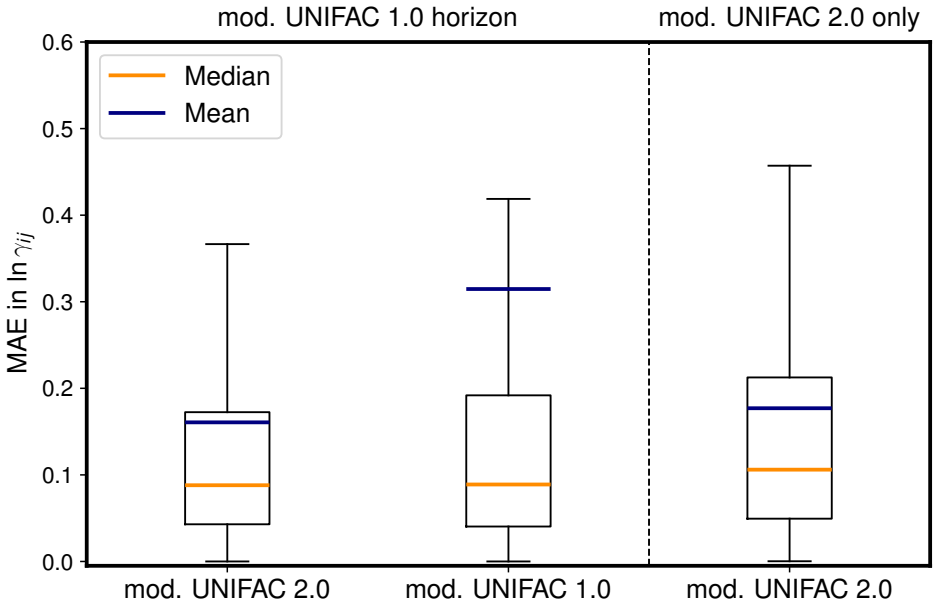


Figure 44: Mean absolute error (MAE) of the predicted $\ln \gamma_i$ with mod. UNIFAC 2.0 and comparison to mod. UNIFAC 1.0 for those mixtures that can also be predicted by the latter model ("mod. UNIFAC 1.0 horizon"). The "mod. UNIFAC 1.0 horizon" comprises 221,639 data points for 16,932 binary mixtures, while an additional 21,618 experimental data points for 4,520 binary mixtures could only be predicted with mod. UNIFAC 2.0 ("mod. UNIFAC 2.0 only"). The boxes represent the interquartile ranges (IQR), and the whiskers extend to the last data points within 1.5 times the IQR from the box edges.

The results in Fig. 44 show an improved prediction accuracy for $\ln \gamma_i$ with mod. UNIFAC 2.0 compared to mod. UNIFAC 1.0 for those mixtures that can be described with both models ("mod. UNIFAC 1.0 horizon"). This is particularly evident concerning the mean of the MAE, which is nearly halved with mod. UNIFAC 2.0, demonstrating the ability of mod. UNIFAC 2.0 to reduce very poorly predicted data points. Regarding the median of the MAE and the interquartile range, mod. UNIFAC 2.0 also shows some improvements compared to mod. UNIFAC 1.0.

These results indicate that using the holistic end-to-end training of mod. UNIFAC 2.0 results in an improved set of pair-interaction parameters compared to the one obtained by the classical sequential fit carried out in the development of mod. UNIFAC 1.0. However, the even more significant advantage of mod. UNIFAC 2.0 is that its parameter set is complete, leading to a much broader applicability. By evaluating the results of mod. UNIFAC 2.0 for those mixtures in the data set that cannot be modeled with mod. UNIFAC 1.0 ("mod. UNIFAC 2.0 only") in Fig. 44, a high prediction accuracy is found. It is similar to that of the results obtained with mod. UNIFAC 1.0 for the mixtures to which this method can be applied.

Fig. 45 shows the results for the prediction of h^{E}, where a similar picture as for the prediction of $\ln \gamma_i$ is seen: an improved performance of mod. UNIFAC 2.0 on the "mod. UNIFAC 1.0 horizon", and still high prediction accuracy on the "mod. UNIFAC 2.0 only" data set, for which mod. UNIFAC 1.0 cannot be applied, is found.

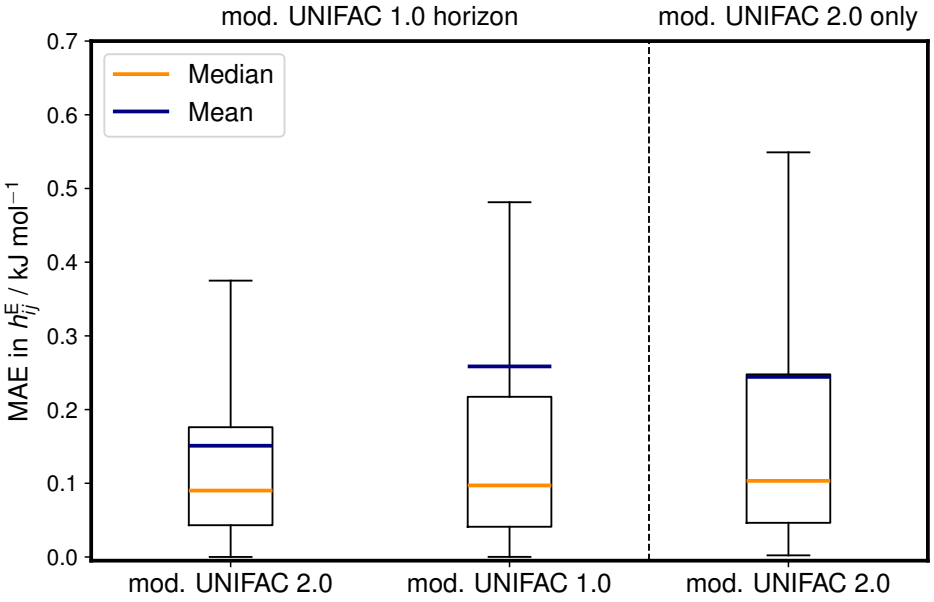


Figure 45: Mean absolute error (MAE) of the predicted h^{E} with mod. UNIFAC 2.0 and comparison to mod. UNIFAC 1.0 for those mixtures that can also be predicted by the latter model ("mod. UNIFAC 1.0 horizon"). The "mod. UNIFAC 1.0 horizon" comprises 239,770 data points for 7,776 binary mixtures, while an additional 19,937 experimental data points for 959 binary mixtures could only be predicted with mod. UNIFAC 2.0 ("mod. UNIFAC 2.0 only"). The boxes represent the interquartile ranges (IQR), and the whiskers extend to the last data points within 1.5 times the IQR from the box edges.

Fig. 46 provides a deeper insight into the overall performance of mod. UNIFAC 2.0 by assigning an MAE for predicting $\ln \gamma_i$ to each pair of main groups, visualized as heatmaps. The shown MAEs are calculated by considering the predictions for all mixtures for which the respective group combination is relevant, with the number of mixtures and data points varying significantly among the pairs of main groups, as detailed in Fig. G.1b of Appendix G. Panel (a) of Fig. 46 shows the MAEs calculated as described above on the complete data set, while panel (b) visualizes improvements (or deteriorations) with mod. UNIFAC 2.0 compared to mod. UNIFAC 1.0 by showing the differences in the MAEs (ΔMAE = $\mathrm{MAE}_{\text{mod. UNIFAC 2.0}} - \mathrm{MAE}_{\text{mod. UNIFAC 1.0}}$) on the "mod. UNIFAC 1.0 horizon". Missing entries indicate that no data were available to compare the given combination of groups.

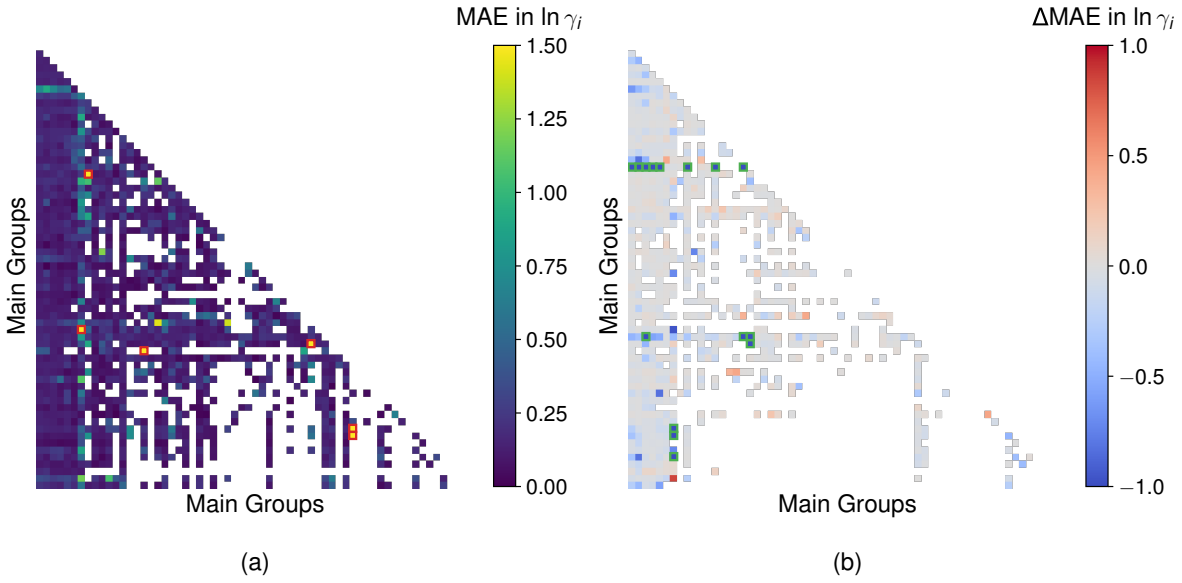


Figure 46: (a) Heatmap of the mean absolute error (MAE) of the predicted $\ln \gamma_i$ with mod. UNIFAC 2.0 calculated for each pair of main groups by considering all data points for which that particular group combination is relevant. Group combinations with an MAE above 1.5 are highlighted by red frames. (b) Difference between the MAE in $\ln \gamma_i$ with mod. UNIFAC 2.0 and the MAE of mod. UNIFAC 1.0 on the "mod UNIFAC 1.0 horizon" (ΔMAE = $\text{MAE}_{\text{mod. UNIFAC 2.0}} - \text{MAE}_{\text{mod. UNIFAC 1.0}}$) for each pair of main groups. Group combinations with a ΔMAE below -1 are highlighted by green frames, indicating the most significant improvements with mod. UNIFAC 2.0. Missing entries indicate that no data were available for the comparison of the given combination of groups.

Fig. 46a highlights the overall strong performance of mod. UNIFAC 2.0, with a small MAE for most group combinations. Note that the prediction for a particular mixture usually requires the consideration of multiple pair-interaction parameters. Hence, the MAEs in Fig. 46a, although assigned to specific group combinations, cannot be attributed to imperfections of the respective pair-interaction parameters alone, but are also affected by other pair-interaction parameters. However, despite this complexity, a clear trend can be observed. For instance, mixtures containing water (main group 7) apparently represent a particular challenge, likely because of the unique properties of water due to strong hydrogen bonding and polarity. While most group combinations yield an MAE below 0.14, which is a very good result, a few show high prediction errors. The three group combinations with an MAE greater than 2.0 are cases based on extremely small test sets, each consisting of a single binary mixture with ten or fewer data points. This suggests that the higher errors may also be due to limited data, and caution should be taken not to over-interpret these results.

Fig. 46b shows that mod. UNIFAC 2.0 outperforms mod. UNIFAC 1.0 for most group combinations. It significantly improves the results for 461 interaction parameters, with

a mean ΔMAE of -0.31, whereas for the 267 combinations mod. UNIFAC 1.0 yields better results; however, the deterioration is typically only minor with a mean ΔMAE of only 0.05. Notable improvements are observed for parameters involving main groups 7 ("H2O"), 18 ("PYRIDINE"), and 42 ("CY-CH2"), with ten group combinations showing extremely high MAE reductions with $\Delta\text{MAE} < -4$. In addition, parameters involving the most common group, main group 1 ("CH3"), also show significant improvements. For example, the mean MAE specific for the pair-interaction parameter between main group 1 ("CH3") and main group 7 ("H2O"), known to be poorly fitted in mod. UNIFAC 1.0, is nearly halved, from 1.35 to 0.71.

Fig. 47 shows an example of the practical application of mod. UNIFAC 2.0. It is used to predict vapor-liquid phase equilibria for binary mixtures, a critical task in chemical engineering. Six typical examples are shown, covering a range of phase behaviors from near-ideal mixtures to those with significant deviations, including low- and high-boiling azeotropes. All shown mixtures are part of the "mod. UNIFAC 2.0 only" set, i.e., they cannot be modeled with mod. UNIFAC 1.0. The predictions show an excellent agreement with the experimental data, underscoring the method's utility for modeling complex phase behavior and making it a valuable tool for a wide range of industrial processes, from distillation to solvent recovery.

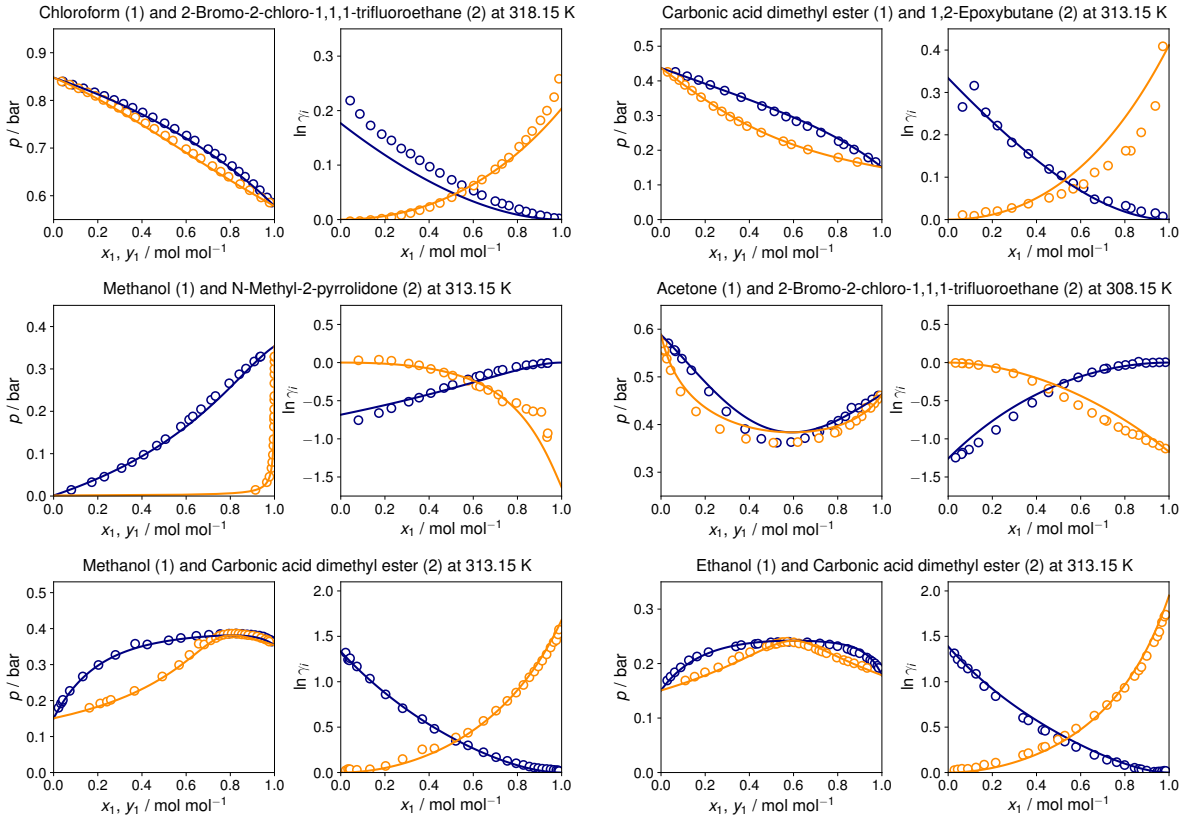


Figure 47: Prediction of $\ln \gamma_i$ and isothermal vapor–liquid phase diagrams for binary mixtures with mod. UNIFAC 2.0 (lines) and comparison to experimental data from the DDB (symbols). Mod. UNIFAC 1.0 is not applicable to the mixtures shown.

Fig. 48 demonstrates the ability of mod. UNIFAC 2.0 to predict excess enthalpies h^{E} in binary mixtures. The figure presents six representative examples from the "mod. UNIFAC 2.0 only" data set, i.e., mixtures for which mod. UNIFAC 1.0 is not applicable, cf. Fig. 45. The mixtures have been selected to highlight a variety of behaviors, ranging from nearly ideal to strongly non-ideal systems with both positive and negative deviations. The predicted excess enthalpy curves (solid lines) align closely with the experimental data (open circles), demonstrating the model's ability to accurately capture both the magnitude and the trend of h^{E} across different mixtures.

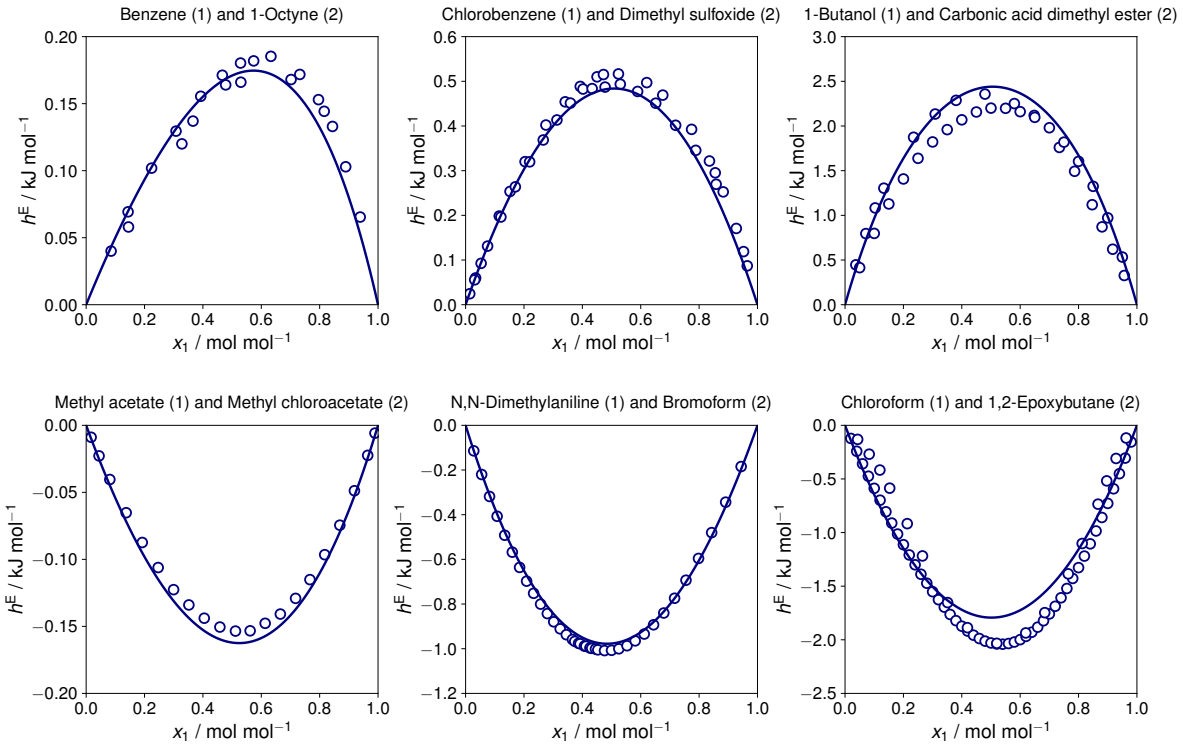


Figure 48: Prediction of excess enthalpies h^{E} at 298.15 K for binary mixtures with mod. UNIFAC 2.0 (lines) and comparison to experimental data from the DDB (symbols). Mod. UNIFAC 1.0 is not applicable to the mixtures shown.

Furthermore, since mod. UNIFAC 2.0 is based on pairwise interactions between the structural groups occurring in the mixture, for any number of components, it allows for straightforward predictions of the properties of multi-component mixtures. Fig. G.3 in Appendix G shows examples for modeling ternary mixtures with mod. UNIFAC 2.0, which demonstrate its high predictive performance although being trained only on binary data.

5.2.2.3.2 Extrapolation to Unseen Components

To study the ability of mod. UNIFAC 2.0 to extrapolate to mixtures involving components for which no mixture data were used in the training (termed "unseen components" in the following for simplicity), a test was carried out in which 100 components were selected, and all data points containing any of these components were withheld from the training set and used only for testing the predictions. This exclusion resulted in a test set comprising 34,107 data points (20,912 for $\ln \gamma_i$ and 13,195 for h^{E}), covering 1,865 different binary mixtures. Fig. 49 shows the results for the prediction of $\ln \gamma_i$ for this test set, again represented as box plots of the mixture-specific MAE. Results from mod. UNIFAC 1.0 are also shown for comparison.

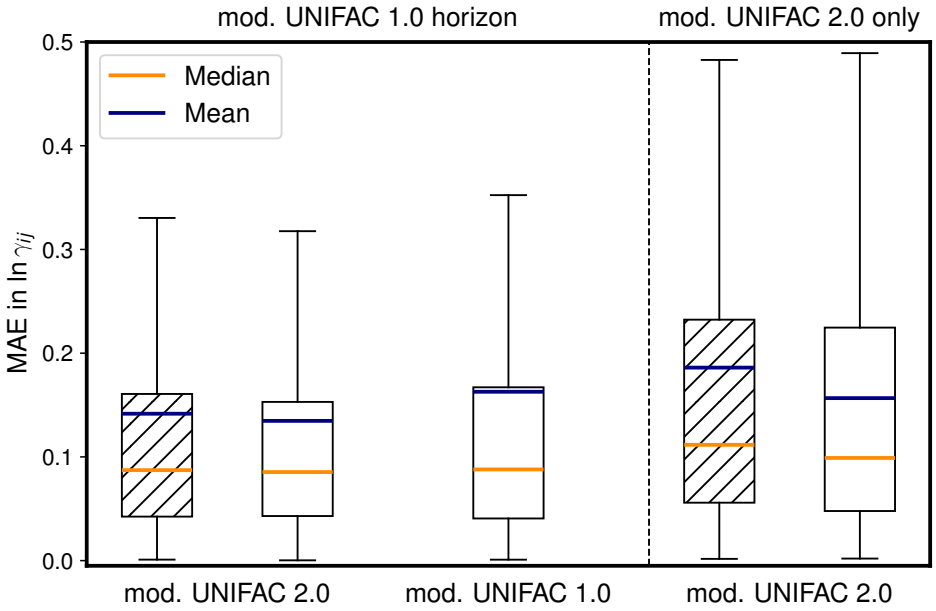


Figure 49: Mean absolute error (MAE) of the predicted $\ln \gamma_i$ of mixtures containing unseen components with mod. UNIFAC 2.0 (shaded boxes). For comparison, the results of mod. UNIFAC 2.0 trained on all experimental data and mod. UNIFAC 1.0 are also shown (plain boxes). The "mod. UNIFAC 1.0 horizon" comprises 19,015 data points for 1,254 binary mixtures, while an additional 1,897 experimental data points for 280 binary mixtures could only be predicted with mod. UNIFAC 2.0 ("mod. UNIFAC 2.0 only"). The boxes represent the interquartile ranges (IQR), and the whiskers extend to the last data points within 1.5 times the IQR from the box edges.

Fig. 49 shows that the predictive accuracy of mod. UNIFAC 2.0 for mixtures with components that were excluded from the training set (shaded boxes) is only narrowly lower than when the model is trained on the entire database (plain boxes). This consistency across both the "mod. UNIFAC 1.0 horizon" and "mod. UNIFAC 2.0 only" data sets underscores the robustness of the hybrid approach. Moreover, even on the test data, mod. UNIFAC 2.0 outperforms mod. UNIFAC 1.0 on the "mod. UNIFAC 1.0 horizon", which is noteworthy given that mod. UNIFAC 1.0 was likely trained on many of these test data points, as discussed earlier. On the "mod. UNIFAC 2.0 only" data set, mod. UNIFAC 2.0 shows slightly reduced predictive accuracy but still maintains strong performance, while mod. UNIFAC 1.0 is not applicable. Overall, these results highlight the predictive power of mod. UNIFAC 2.0. Similar trends were observed for the prediction of h^{E}, as shown in Fig. G.4 in Appendix G.

5.2.2.3.3 Extrapolation to Unseen Pair-Interaction Parameters

Another, even more challenging, test to assess mod. UNIFAC 2.0's predictive capacities is to test its ability to extrapolate to unseen pair interactions. For such a test, 100

combinations of main groups have been randomly selected, and all experimental data for mixtures for which the respective main group combinations are relevant have been withheld from the training. For each of these 100 combinations, an individual test set was created from the withheld data, while all other available data were used to train mod. UNIFAC 2.0. The number of data points and binary mixtures for the 100 test sets, as well as individual error scores, are given in Tables G.1 (for $\ln \gamma_i$) and G.2 (for h^{E}) in Appendix G.

Fig. 50 shows the results of predicting $\ln \gamma_i$ with mod. UNIFAC 2.0 from this challenging test by summarizing the MAEs for the 100 test sets in a box plot. For comparison, the performance of mod. UNIFAC 2.0 trained on all experimental data and the results of mod. UNIFAC 1.0 (on the "mod. UNIFAC 1.0 horizon") are included. Similar results were obtained for h^{E} and are summarized in Fig. G.6 in Appendix G.

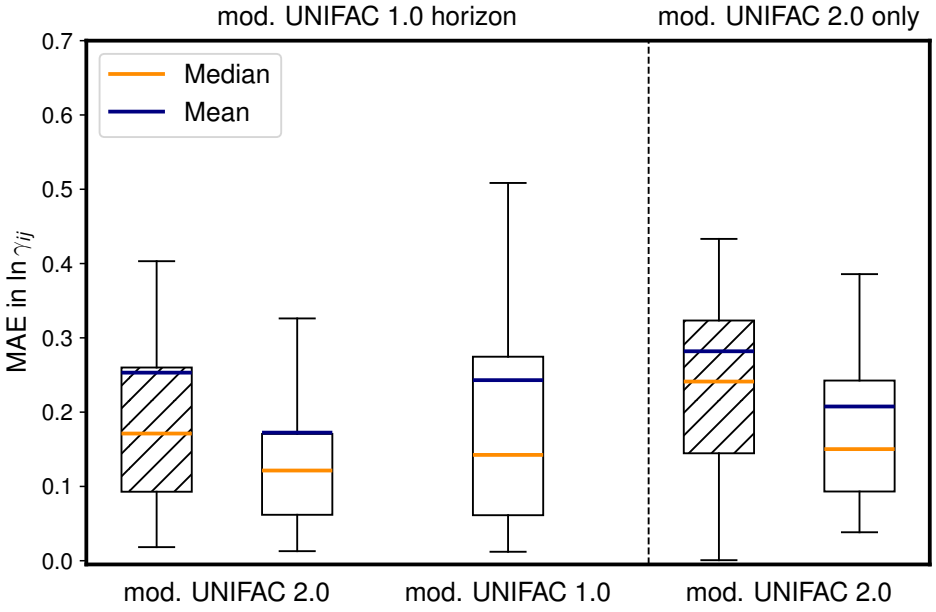


Figure 50: Mean absolute error (MAE) of the predicted $\ln \gamma_i$ with mod. UNIFAC 2.0 for 100 test sets, where all data points for which a specific main group combination is relevant were withheld during training (shaded boxes); cf. Table G.1 in Appendix G for numerical results. The results of mod. UNIFAC 2.0 trained on all experimental data and mod. UNIFAC 1.0 are shown for comparison (plain boxes). The boxes represent the interquartile ranges (IQR), and the whiskers extend to the last data points within 1.5 times the IQR from the box edges.

The results on the "mod. UNIFAC 1.0 horizon" demonstrate that mod. UNIFAC 2.0, even when predicting truly unseen pair-interaction parameters, which could not directly be fitted to the training data, achieves a performance comparable to mod. UNIFAC 1.0 with parameters that were likely fitted directly to the respective experimental data.

Comparing mod. UNIFAC 2.0's predictions for unseen pair interactions (shaded boxes) with those trained on the entire database (plain boxes) reveals a decrease in accuracy, as expected. However, the differences are modest, highlighting the robustness and reliability of mod. UNIFAC 2.0 even in this extremely challenging test.

These tests emphasize the potential of mod. UNIFAC 2.0 not only to broaden the applicability of this group-contribution method but also to improve its prediction accuracy significantly. Unlike mod. UNIFAC 1.0, which is constrained by its limited parameter tables obtained from sequential fitting, mod. UNIFAC 2.0 excels in both scope and performance, making it a robust tool for predicting activity coefficients across a wide range of mixtures. The superior accuracy demonstrated on the shared horizon confirms that mod. UNIFAC 2.0 is not just a complementary option when mod. UNIFAC 1.0 fails but a strong candidate to become the new standard.

Its ease of implementation sets mod. UNIFAC 2.0 apart from other ML-based or hybrid models combining ML with physical modeling. Users can seamlessly adopt mod. UNIFAC 2.0 by simply replacing the original parameter tables in their existing process simulators (or similar software), in which mod. UNIFAC will most likely be implemented, with the completed parameter tables provided in Ref. [145]. This way, the tedious implementation of the ML model itself is eliminated, making mod. UNIFAC 2.0 directly accessible for practical applications.

5.2.2.4 Conclusions

Mod. UNIFAC [13] is currently the industrial standard for predicting activity coefficients and is implemented in basically all process simulation software packages. It is also widely used in academia and is the workhorse for calculating phase equilibria with liquid phases, such as vapor-liquid equilibria (VLE), liquid-liquid equilibria (LLE), and solid-liquid equilibria (SLE). Due to temperature-dependent parameters, it is more flexible than the original UNIFAC model [12] and often delivers better results. Furthermore, mod. UNIFAC often gives better results than competing excess Gibbs energy models based on quantum-mechanical calculations of energetic contributions, such as COSMO-RS [34, 43, 44] and COSMO-SAC-dsp [47].

However, mod. UNIFAC has several important drawbacks. Firstly, the last published version stems from 2016 and has therefore been fitted only to data that were available up to then. Hence, the wealth of relevant data measured since then is omitted. More recent updates of mod. UNIFAC are commercial and not publicly available. Secondly, and more importantly, as a group-contribution method, mod. UNIFAC can only be applied to make predictions for a given mixture if all pair-interaction parameters (between all groups into which all components of the mixture are decomposed) are available. If

only a single pair is missing, mod. UNIFAC will not work. The latest public version of mod. UNIFAC has 63 main groups, and, hence, 1953 pairs of groups – but interaction parameters are only available for 756 of these pairs (39%), which considerably limits the method's applicability.

Therefore, mod. UNIFAC 2.0 has been developed, which overcomes these drawbacks: It was trained on data for activity coefficients and excess enthalpies published up to 2024 that were taken from the Dortmund Data Bank (DDB). All in all, more than 500,000 data points from 27,035 binary systems were used. The equations and the groups used in mod. UNIFAC 2.0 are exactly the same as in the last published version, called mod. UNIFAC 1.0 here, but the training differs drastically. While the parameters of mod. UNIFAC 1.0 were determined in a sequential approach, without a chance to fill gaps for interactions for which no relevant data were available, mod. UNIFAC 2.0 is trained using a matrix completion method (MCM) by which the entire interaction parameter matrix is filled simultaneously. Consequently, there are no gaps in the mod. UNIFAC 2.0 parameter tables. This leads to an important extension of the applicability of the method. However, not only was the applicability extended, but the accuracy of the predictions was also improved. This was demonstrated in tests in which mod. UNIFAC 2.0 was compared to mod. UNIFAC 1.0: In different studies, data were deliberately excluded from the training and only used for the tests. Even in these tests, mod. UNIFAC 2.0 performed consistently better than mod. UNIFAC 1.0, even though they favor mod. UNIFAC 1.0, as it must be assumed that relevant parts of the test set were used in its training. In-depth studies also reveal significant improvements for technically important classes of mixtures, such as mixtures containing water.

As a method based on the physical concept of pair interactions, mod. UNIFAC 2.0 can be used to predict thermodynamic properties not only for binary mixtures but also for multi-component mixtures. The new model can be seamlessly integrated into existing workflows, as users only need to update the parameter tables in existing implementations. Ultimately, the end-to-end training process of mod. UNIFAC 2.0 allows for straightforward updates as new experimental data become available or for tailoring the model to specific industrial needs. Mod. UNIFAC 2.0 demonstrates how combining machine learning with established physical models can significantly enhance the prediction of thermodynamic properties. Its expanded scope, improved accuracy, and ease of implementation represent a powerful and scalable solution for modern chemical engineering challenges. The complete parameter tables are freely provided in Ref. [145] as .csv files. It is recommended to use mod. UNIFAC 2.0 as the default in all applications where, up to now, the default was mod. UNIFAC 1.0.

6 Conclusions

Knowledge of the thermodynamic properties of mixtures is essential in chemical engineering and related fields. However, due to the vast combinatorial diversity of mixtures and the high cost and effort related to measurements, experimental studies alone cannot cover the full range of compositions and conditions required by industry and research. Consequently, reliable and widely applicable prediction methods are required. In this thesis, a new family of such methods – developed using matrix completion methods (MCMs) from machine learning – is introduced. MCMs are based on the idea that data relevant to thermodynamic properties of mixtures can often be arranged in matrix form and MCMs can be used to complete these matrices even when they are only sparsely populated. MCMs thereby are applied in two ways: to matrices containing thermodynamic property data of binary mixtures and to matrices containing pair-interaction parameters of well established physical group-contribution (GC) methods.

In the present thesis, an MCM for predicting activity coefficients at infinite dilution γ_{ij}^{∞} in binary mixtures has been developed, which is based on a new way to measure the similarity between two components. This similarity measure relies solely on input information readily obtainable from quantum-chemical calculations or standard databases, making it highly versatile. Despite the relatively small experimental database, this similarity-based method (SBM) demonstrates remarkable accuracy and outperforms the established physical benchmark methods modified UNIFAC (Dortmund) [13], COSMO-SAC [46], and COSMO-SAC-dsp [47].

Furthermore, Bayesian MCMs have been developed to directly factorize various thermodynamic properties of binary mixtures, including Henry's law constants H_{ij}, activity coefficients at infinite dilution γ_{ij}^{∞}, and diffusion coefficients at infinite dilution D_{ij}^{∞}. For all considered thermodynamic properties, hybridization strategies that incorporate synthetic data from physical models into the MCM training process were used to improve predictive performance. These hybrid MCMs outperform purely data-driven ones and the physical benchmarks, such as the Predictive Soave-Redlich-Kwong equation-of-state [76] for predicting H_{ij}, modified UNIFAC (Dortmund), COSMO-SAC, and COSMO-SAC-dsp for predicting γ_{ij}^{∞}, and semiempirical methods like SEGWE for predicting D_{ij}^{∞}.

Moreover, the significant limitations of GC methods, the previous gold standard for predicting thermodynamic properties of mixtures, due to incomplete sets of pair-interaction parameters were addressed in the present thesis by embedding Bayesian MCMs in the framework of physical GC methods. In the resulting models, the MCMs provide a complete set of pair-interaction parameters and, therefore, substantially extend the applicability of the GC methods. First, an MCM to complete the group-interaction parameter set of UNIFAC [12] was developed that was trained solely on pseudo-data generated with UNIFAC based on the original parameterization. This method already achieves a similar performance as the original model while substantially extending its scope. By developing another method of this type that was trained directly on experimental data in an end-to-end manner, a consistently better performance than the original UNIFAC was achieved. Applying this concept to modified UNIFAC (Dortmund) yielded comparable improvements, again enhancing both accuracy and applicability. These upgraded UNIFAC models can be integrated seamlessly into existing workflows, requiring only parameter table updates. Moreover, as new experimental data become available, the end-to-end training procedure allows for swift model updates, making the approach flexible and adaptable to evolving industrial requirements.

Opportunities for future work include leveraging the insights from this work to develop targeted design-of-experiments strategies. Specifically, it has been demonstrated that the sheer amount of data is not sufficient for MCMs to achieve very high prediction accuracy; the training data must contain information on mixtures that are similar to the target mixtures, which can be assessed by the proposed similarity measure.

Furthermore, the strong performance of hybrid MCMs that factorize thermodynamic properties motivates their extension to other conditions and the prediction of other thermodynamic properties in future work. It is interesting that the matrix-completion approach emerges not as a competitor to the established methods, but rather as a complement, which could be an inspiration for future investigations of coupling machine learning approaches with existing physical models to create the next generation of powerful hybrid predictive models.

The hybrid UNIFAC models proposed in the present thesis can be extended in many ways. The incorporation of additional training data on other mixture properties, such as liquid-liquid equilibria, would be of great interest. In addition, overcoming the limitations imposed by public tables that define component decompositions remains an important goal. The introduction of improved sets of structural groups and an automated framework for decomposing components into these groups would greatly extend the applicability of these methods.

Apart from the UNIFAC models, most physical models of thermodynamic properties of mixtures are based on the concept of pair interactions. Hence, MCMs can be embedded

in all these models to predict the underlying pair interactions. This allows to obtain parameters in cases where not sufficient experimental data for a conventional parameterization are available. The results from the present work have also demonstrated that the consistent simultaneous approach for the parameterization of the models using an end-to-end training on large data sets clearly outperforms the traditional stepwise procedure. Such advances would mark a significant step toward the creation of universally adaptable and continuously updatable predictive models of thermodynamic properties that support the design and optimization of complex chemical processes.

Literature

[1] F. Jirasek, H. Hasse: Perspective: Machine learning of thermophysical properties, Fluid Phase Equilibria 549 (2021) 113206. DOI: 10.1016/j.fluid.2021.113206.

[2] F. Jirasek, H. Hasse: Combining machine learning with physical knowledge in thermodynamic modeling of fluid mixtures, Annual review of chemical and biomolecular engineering 14 (2023) 31–51. DOI: 10.1146/annurev-chembioeng-092220-025342.

[3] A. M. Schweidtmann, E. Esche, A. Fischer, M. Kloft, J.-U. Repke, S. Sager, A. Mitsos: Machine learning in chemical engineering: A perspective, Chemie Ingenieur Technik 93 (2021) 2029–2039. DOI: 10.1002/cite.202100083.

[4] A. Ramlatchan, M. Yang, Q. Liu, M. Li, J. Wang, Y. Li: A survey of matrix completion methods for recommendation systems, Big Data Mining and Analytics 1 (2018) 308–323. DOI: 10.26599/BDMA.2018.9020008.

[5] C. Teflioudi, F. Makari, R. Gemulla: Distributed matrix completion, in: 2012 IEEE 12th International Conference on Data Mining, 2012, pp. 655–664. DOI: 10.1109/ICDM.2012.120.

[6] J. Bennett, S. Lanning: The netflix prize, in: A. SIGKDD, Netflix (Eds.), Proceedings of the KDD Cup and Workshop, ACM, 2007, p. 35.

[7] Z. Chen, S. Wang: A review on matrix completion for recommender systems, Knowledge and Information Systems 64 (2022) 1–34. DOI: 10.1007/s10115-021-01629-6.

[8] F. Jirasek, R. A. S. Alves, J. Damay, R. A. Vandermeulen, R. Bamler, M. Bortz, S. Mandt, M. Kloft, H. Hasse: Machine learning in thermodynamics: Prediction of activity coefficients by matrix completion, The journal of physical chemistry letters 11 (2020) 981–985. DOI: 10.1021/acs.jpclett.9b03657.

[9] F. Jirasek, R. Bamler, S. Mandt: Hybridizing physical and data-driven prediction methods for physicochemical properties, Chemical Communications 56 (2020) 12407–12410. DOI: 10.1039/D0CC05258B.

[10] J. Damay, F. Jirasek, M. Kloft, M. Bortz, H. Hasse: Predicting activity coefficients at infinite dilution for varying temperatures by matrix completion, Industrial & Engineering Chemistry Research 60 (2021) 14564–14578. DOI: 10.1021/acs.iecr.1c02039.

[11] J. Damay, G. Ryzhakov, F. Jirasek, H. Hasse, I. Oseledets, M. Bortz: Predicting temperature–dependent activity coefficients at infinite dilution using tensor completion, Chemie Ingenieur Technik 95 (2023) 1061–1069. DOI: 10.1002/cite.202200230.

[12] R. Wittig, J. Lohmann, J. Gmehling: Vapor–liquid equilibria by UNIFAC group contribution. 6. revision and extension, Industrial & Engineering Chemistry Research 42 (2003) 183–188. DOI: 10.1021/ie0205061.

[13] D. Constantinescu, J. Gmehling: Further development of modified UNIFAC (Dortmund): Revision and extension 6, Journal of Chemical & Engineering Data 61 (2016) 2738–2748. DOI: 10.1021/acs.jced.6b00136.

[14] Y. Koren, R. Bell, C. Volinsky: Matrix factorization techniques for recommender systems, Computer 42 (2009) 30–37. DOI: 10.1109/mc.2009.263.

[15] D. M. Blei, A. Kucukelbir, J. D. McAuliffe: Variational inference: A review for statisticians, Journal of the American Statistical Association 112 (2017) 859–877. DOI: 10.1080/01621459.2017.1285773.

[16] L. Tierney: Markov chains for exploring posterior distributions, the Annals of Statistics (1994) 1701–1728. DOI: 10.1214/aos/1176325750.

[17] Q. Jones, Galin L; Qin: Markov chain monte carlo in practice, Annual Review of Statistics and Its Application 9 (2022) 557–578. DOI: 10.1146/annurev-statistics-040220-090158.

[18] C. Zhang, J. Butepage, H. Kjellstrom, S. Mandt: Advances in variational inference, IEEE Trans. Pattern Anal. Mach. Intell. 41 (2019) 2008–2026. DOI: 10.1109/TPAMI.2018.2889774.

[19] A. Kucukelbir, D. Tran, R. Ranganath, A. Gelman, D. M. Blei: Automatic differentiation variational inference, Journal of Machine Learning Research 18 (2017) 1–45.

[20] B. Carpenter, A. Gelman, M. D. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. A. Brubaker, J. Guo, P. Li, A. Riddell: Stan: A probabilistic programming language, Grantee Submission 76 (2017) 1–32. DOI: 10.18637/jss.v076.i01.

[21] The MathWorks Inc.: Matlab r2019b, 2019. URL: www.mathworks.com.

[22] The MathWorks Inc.: Matlab r2021b, 2021. URL: www.mathworks.com.

[23] The MathWorks Inc.: Matlab r2022b, 2022. URL: www.mathworks.com.

[24] E. Bingham, J. P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletsos, R. Singh, P. Szerlip, P. Horsfall, N. D. Goodman: Pyro: Deep universal probabilistic programming, Journal of Machine Learning Research (2018).

[25] D. P. Kingma, J. Ba: Adam: A method for stochastic optimization, 2014. URL: http://arxiv.org/pdf/1412.6980.pdf.

[26] C. C. Aggarwal: Recommender systems: The textbook, first edition ed., Springer, Cham, 2016. DOI: 10.1007/978-3-319-29659-3.

[27] S. K. Raghuwanshi, R. K. Pateriya: Collaborative filtering techniques in recommendation systems, in: R. K. Shukla, J. Agrawal, S. Sharma, G. Singh Tomer (Eds.), Data, Engineering and Applications: Volume 1, Springer, Singapore, 2019, pp. 11–21. DOI: 10.1007/978-981-13-6347-4.

[28] N. Nikolova, J. Jaworska: Approaches to measure chemical similarity – a review, QSAR & Combinatorial Science 22 (2003) 1006–1026. DOI: 10.1002/qsar.200330831.

[29] D. Stumpfe, J. Bajorath: Similarity searching, WIREs Computational Molecular Science 1 (2011) 260–282. DOI: 10.1002/wcms.23.

[30] D. R. Flower: On the properties of bit string-based measures of chemical similarity, Journal of Chemical Information and Computer Sciences 38 (1998) 379–386. DOI: 10.1021/ci970437z.

[31] M. A. Fligner, J. S. Verducci, P. E. Blower: A modification of the jaccard–tanimoto similarity index for diverse selection of chemical compounds using binary strings, Technometrics 44 (2002) 110–119. DOI: 10.1198/004017002317375064.

[32] D. Bajusz, A. Rácz, K. Héberger: Why is tanimoto index an appropriate choice for fingerprint-based similarity calculations?, Journal of Cheminformatics 7 (2015) 20. DOI: 10.1186/s13321-015-0069-3.

[33] J. W. Raymond, P. Willett: Effectiveness of graph-based and fingerprint-based similarity measures for virtual screening of 2D chemical structure databases, Journal of Computer-Aided Molecular Design 16 (2002) 59–71. DOI: 10.1023/A:1016387816342.

[34] A. Klamt: Conductor-like screening model for real solvents: A new approach to the quantitative calculation of solvation phenomena, The Journal of Physical Chemistry 99 (1995) 2224–2235. DOI: 10.1021/j100007a062.

[35] M. Thormann, A. Klamt, K. Wichmann: COSMOsim3D: 3D-similarity and alignment based on COSMO polarization charge densities, Journal of chemical information and modeling 52 (2012) 2149–2156. DOI: 10.1021/ci300205p.

[36] M. Thormann, N. Traube, N. Yehia, R. Koestler, G. Galabova, N. MacAulay, T. L. Toft-Bertelsen: Toward new AQP4 inhibitors: ORI-TRN-002, International Journal of Molecular Sciences 25 (2024) 924. DOI: 10.3390/ijms25020924.

[37] T. Brouwer, B. Schuur: Model performances evaluated for infinite dilution activity coefficients prediction at 298.15 K, Industrial & Engineering Chemistry Research 58 (2019) 8903–8914. DOI: 10.1021/acs.iecr.9b00727.

[38] DDBST - Dortmund Data Bank Software & Separation Technology GmbH: Dortmund Data Bank, 2023. URL: www.ddbst.com.

[39] H. Orbey, S. I. Sandler: Relative measurements of activity coefficients at infinite dilution by gas chromatography, Industrial & Engineering Chemistry Research 30 (1991) 2006–2011. DOI: 10.1021/ie00056a051.

[40] K. Kojima, S. Zhang, T. Hiaki: Measuring methods of infinite dilution activity coefficients and a database for systems including water, Fluid Phase Equilibria 131 (1997) 145–179. DOI: 10.1016/S0378-3812(96)03210-4.

[41] A. Fredenslund, R. L. Jones, J. M. Prausnitz: Group-contribution estimation of activity coefficients in nonideal liquid mixtures, AIChE Journal 21 (1975) 1086–1099. DOI: 10.1002/aic.690210607.

[42] U. Weidlich, J. Gmehling: A modified UNIFAC model. 1. prediction of VLE, h^{E}, and γ^{∞}, Industrial & Engineering Chemistry Research 26 (1987) 1372–1381. DOI: 10.1021/ie00067a018.

[43] A. Klamt, F. Eckert: COSMO-RS: a novel and efficient method for the a priori prediction of thermophysical data of liquids, Fluid Phase Equilibria 172 (2000) 43–72. DOI: 10.1016/s0378-3812(00)00357-5.

[44] A. Klamt: COSMO-RS: From quantum chemistry to fluid phase thermodynamics and drug design, 1st ed. ed., Elsevier, Amsterdam, 2005.

[45] S.-T. Lin, S. I. Sandler: A priori phase equilibrium prediction from a segment contribution solvation model, Industrial & Engineering Chemistry Research 41 (2002) 899–913. DOI: 10.1021/ie001047w.

[46] C.-M. Hsieh, S. I. Sandler, S.-T. Lin: Improvements of COSMO-SAC for vapor–liquid and liquid–liquid equilibrium predictions, Fluid Phase Equilibria 297 (2010) 90–97. DOI: 10.1016/j.fluid.2010.06.011.

[47] C.-M. Hsieh, S.-T. Lin, J. Vrabec: Considering the dispersive interactions in the COSMO-SAC model for more accurate predictions of fluid phase behavior, Fluid Phase Equilibria 367 (2014) 109–116. DOI: 10.1016/j.fluid.2014.01.032.

[48] E. I. Sanchez Medina, S. Linke, M. Stoll, K. Sundmacher: Graph neural networks for the prediction of infinite dilution activity coefficients, Digital Discovery 1 (2022) 216–225. DOI: 10.1039/D1DD00037C.

[49] B. Winter, C. Winter, J. Schilling, A. Bardow: A smile is all you need: predicting limiting activity coefficients from SMILES with natural language processing, Digital Discovery 1 (2022) 859–869. DOI: 10.1039/d2dd00058j.

[50] B. Winter, C. Winter, T. Esper, J. Schilling, A. Bardow: SPT-NRTL: A physics-guided machine learning model to predict thermodynamically consistent activity coefficients, Fluid Phase Equilibria 568 (2023) 113731. DOI: 10.1016/j.fluid.2023.113731.

[51] F. Jirasek, N. Hayer, R. Abbas, B. Schmid, H. Hasse: Prediction of parameters of group contribution models of mixtures by matrix completion, Physical chemistry chemical physics : PCCP 25 (2023) 1054–1062. DOI: 10.1039/d2cp04478a.

[52] J. G. Rittig, K. C. Felton, A. A. Lapkin, A. Mitsos: Gibbs–Duhem-informed neural networks for binary activity coefficient prediction, Digital Discovery 2 (2023) 1752–1767. DOI: 10.1039/D3DD00103B.

[53] T. Specht, M. Nagda, S. Fellenz, S. Mandt, H. Hasse, F. Jirasek: HANNA: Hard-constraint neural network for consistent activity coefficient prediction, Chemical science (2024). DOI: 10.1039/D4SC05115G.

[54] N. Hayer, T. Wendel, S. Mandt, H. Hasse, F. Jirasek: Advancing thermodynamic group-contribution methods by machine learning: UNIFAC 2.0, Chemical Engineering Journal 504 (2024) 158667. DOI: 10.1016/j.cej.2024.158667.

[55] I. H. Bell, E. Mickoleit, C.-M. Hsieh, S.-T. Lin, J. Vrabec, C. Breitkopf, A. Jäger: A benchmark open-source implementation of COSMO-SAC, Journal of Chemical Theory and Computation 16 (2020) 2635–2646. DOI: 10.1021/acs.jctc.9b01016.

[56] T. Hastie, R. Tibshirani, J. H. Friedman: The elements of statistical learning: Data mining, inference, and prediction, Springer Series in Statistics, second edition ed., Springer, New York, NY, 2017. DOI: 10.1007/978-0-387-84858-7.

[57] D. Gond, J.-T. Sohns, H. Leitte, H. Hasse, F. Jirasek: Hierarchical matrix completion for the prediction of properties of binary mixtures, arXiv preprint, 2024. URL: http://arxiv.org/pdf/2410.06060v1, arXiv:2410.06060.

[58] E. J. Candès, B. Recht: Exact matrix completion via convex optimization, Foundations of Computational Mathematics 9 (2009) 717–772. DOI: 10.1007/s10208-009-9045-5.

[59] R. Parhizkar, A. Karbasi, S. Oh, M. Vetterli: Calibration using matrix completion with application to ultrasound tomography, IEEE Transactions on Signal Processing 61 (2013) 4923–4933. DOI: 10.1109/TSP.2013.2272925.

[60] A. Ledent, R. Alves, M. Kloft: Orthogonal inductive matrix completion, arXiv e-prints (2020) arXiv:2004.01653.

[61] R. Salakhutdinov, A. Mnih: Bayesian probabilistic matrix factorization using Markov chain Monte Carlo, in: W. Cohen (Ed.), Proceedings of the 25th international conference on Machine learning, ACM, New York, NY, 2008, pp. 880–887. DOI: 10.1145/1390156.1390267.

[62] S. Kim, Yong-Deok; Choi: Scalable variational Bayesian matrix factorization with side information, Artificial Intelligence and Statistics (2014) 493–502.

[63] M.-A. Ahmadi, B. Pouladi, Y. Javvi, S. Alfkhani, R. Soleimani: Connectionist technique estimates H_2S solubility in ionic liquids through a low parameter approach, The Journal of Supercritical Fluids 97 (2015) 81–87. DOI: 10.1016/j.supflu.2014.11.009.

[64] N. J. English, D. G. Carroll: Prediction of Henry's law constants by a quantitative structure property relationship and neural networks, Journal of Chemical Information and Computer Sciences 41 (2001) 1150–1161. DOI: 10.1021/ci010361d.

[65] D. R. O'Loughlin, N. J. English: Prediction of Henry's law constants via group-specific quantitative structure property relationships, Chemosphere 127 (2015) 1–9. DOI: 10.1016/j.chemosphere.2014.11.065.

[66] Z. Wang, Y. Su, S. Jin, W. Shen, J. Ren, X. Zhang, J. H. Clark: A novel unambiguous strategy of molecular feature extraction in machine learning assisted predictive models for environmental properties, Green Chemistry 22 (2020) 3867–3876. DOI: 10.1039/D0GC01122C.

[67] A. R. Katritzky, M. Kuanar, S. Slavov, C. D. Hall, M. Karelson, I. Kahn, D. A. Dobchev: Quantitative correlation of physical and chemical properties with chemical structure: Utility for prediction, Chemical Reviews 110 (2010) 5714–5789. DOI: 10.1021/cr900238d.

[68] M. Goodarzi, E. V. Ortiz, L. d. S. Coelho, P. R. Duchowicz: Linear and non-linear relationships mapping the Henry's law parameters of organic pesticides, Atmospheric Environment 44 (2010) 3179–3186. DOI: 10.1016/j.atmosenv.2010.05.025.

[69] P. R. Duchowicz, J. F. Aranda, D. E. Bacelo, S. E. Fioressi: QSPR study of the Henry's law constant for heterogeneous compounds, Chemical Engineering Research and Design 154 (2020) 115–121. DOI: 10.1016/j.cherd.2019.12.009.

[70] D. Ghaslani, Z. Eshaghi Gorji, A. Ebrahimpoor Gorji, S. Riahi: Descriptive and predictive models for Henry's law constant of CO_2 in ionic liquids: A QSPR study, Chemical Engineering Research and Design 120 (2017) 15–25. DOI: 10.1016/j.cherd.2016.12.020.

[71] H. Li, X. Wang, T. Yi, Z. Xu, X. Liu: Prediction of Henry's law constants for organic compounds using multilayer feedforward neural networks based on linear salvation energy relationship, Journal of Chemical and Pharmaceutical Research 6 (2014) 1557–1564.

[72] J. Gmehling, B. Kolbe, M. Kleiber, J. R. Rarey: Chemical thermodynamics for process simulation, Wiley-VCH-Verl., Weinheim, 2012.

[73] S. Dahl, M. L. Michelsen: High-pressure vapor-liquid equilibrium with a UNIFAC-based equation of state, AIChE Journal 36 (1990) 1829–1836. DOI: 10.1002/aic.690361207.

[74] N. C. Patel, V. Abovsky, S. Watanasiri: Calculation of vapor–liquid equilibria for a 10-component system: comparison of EOS, EOS–G^E and G^E–Henry's law models, Fluid Phase Equilibria 185 (2001) 397–405. DOI: 10.1016/S0378-3812(01)00498-8.

[75] M.-J. Huron, J. Vidal: New mixing rules in simple equations of state for representing vapour-liquid equilibria of strongly non-ideal mixtures, Fluid Phase Equilibria 3 (1979) 255–271. DOI: 10.1016/0378-3812(79)80001-1.

[76] S. Horstmann, A. Jabłoniec, J. Krafczyk, K. Fischer, J. Gmehling: PSRK group contribution equation of state: comprehensive revision and extension IV, including

critical constants and α-function parameters for 1000 components, Fluid Phase Equilibria 227 (2005) 157–164. DOI: 10.1016/j.fluid.2004.11.002.

[77] T. Holderbaum, J. Gmehling: PSRK: A group contribution equation of state based on UNIFAC, Fluid Phase Equilibria 70 (1991) 251–265. DOI: 10.1016/0378-3812(91)85038-V.

[78] G. Soave: Equilibrium constants from a modified Redlich-Kwong equation of state, Chemical Engineering Science 27 (1972) 1197–1203. DOI: 10.1016/0009-2509(72)80096-4.

[79] S. Horstmann, K. Fischer, J. Gmehling: PSRK group contribution equation of state: revision and extension III, Fluid Phase Equilibria 167 (2000) 173–186. DOI: 10.1016/s0378-3812(99)00333-7.

[80] Q. Yang, C. Zhong: A modified PSRK model for the prediction of the vapor-liquid equilibria of asymmetric systems, Fluid phase equilibria 192 (2001) 103–120. DOI: 10.1016/S0378-3812(01)00629-X.

[81] DDBST - Dortmund Data Bank Software & Separation Technology GmbH: Dortmund Data Bank, 2019. URL: www.ddbst.com.

[82] N. Hayer, F. Jirasek, H. Hasse: Prediction of Henry's law constants by matrix completion, AIChE Journal 68 (2022) e17753. DOI: 10.1002/aic.17753.

[83] O. Großmann, D. Bellaire, N. Hayer, F. Jirasek, H. Hasse: Database for liquid phase diffusion coefficients at infinite dilution at 298 K and matrix completion methods for their prediction, Digital Discovery 1 (2022) 886–897. DOI: 10.1039/D2DD00073C.

[84] F. Jirasek, R. Bamler, S. Fellenz, M. Bortz, M. Kloft, S. Mandt, H. Hasse: Making thermodynamic models of mixtures predictive by machine learning: matrix completion of pair interactions, Chemical science 13 (2022) 4854–4862. DOI: 10.1039/d1sc07210b.

[85] M. Hoffmann, N. Hayer, M. Kohns, F. Jirasek, H. Hasse: Prediction of pair interactions in mixtures by matrix completion, Physical Chemistry Chemical Physics 26 (2024) 19390–19397. DOI: 10.1039/D4CP01492H.

[86] DDBST - Dortmund Data Bank Software & Separation Technology GmbH: Dortmund Data Bank, 2021. URL: www.ddbst.com.

[87] N. Hayer, H. Hasse, F. Jirasek: Prediction of temperature-dependent Henry's law constants by matrix completion, The Journal of Physical Chemistry B 129 (2024) 409–416. DOI: 10.1021/acs.jpcb.4c07196.

[88] M. Görgényi, J. Dewulf, H. Van Langenhove: Temperature dependence of Henry's law constant in an extended temperature range, Chemosphere 7 (2002) 757–762. DOI: 10.1016/S0045-6535(02)00131-5.

[89] K. C. Felton, H. Ben-Safar, A. A. Alexei: DeepGamma: A deep learning model for activity coefficient prediction, in: 1st Annual AAAI Workshop on AI to Accelerate Science and Engineering (AI2ASE), 2022.

[90] E. I. Sanchez Medina, S. Linke, M. Stoll, K. Sundmacher: Gibbs–Helmholtz graph neural network: capturing the temperature dependency of activity coefficients at infinite dilution, Digital Discovery 2 (2023) 781–798. DOI: 10.1039/D2DD00142J.

[91] A. Vignes: Diffusion in binary solutions. variation of diffusion coefficient with composition, Industrial & Engineering Chemistry Fundamentals 5 (1966) 189–199. DOI: 10.1021/i160018a007.

[92] H. A. Kooijman, R. Taylor: Estimation of diffusion coefficients in multicomponent liquid systems, Industrial & Engineering Chemistry Research 30 (1991) 1217–1222. DOI: 10.1021/ie00054a023.

[93] R. Taylor, R. Krishna: Multicomponent Mass Transfer, Wiley series in chemical engineering, Wiley, New York, 1993.

[94] C. R. Wilke, P. Chang: Correlation of diffusion coefficients in dilute solutions, AIChE Journal 1 (1955) 264–270. DOI: 10.1002/aic.690010222.

[95] K. A. Reddy, L. K. Doraiswamy: Estimating liquid diffusivity, Industrial & Engineering Chemistry Fundamentals 6 (1967) 77–79. DOI: 10.1021/i160021a012.

[96] M. T. Tyn, W. F. Calus: Diffusion coefficients in dilute binary liquid mixtures, Journal of Chemical & Engineering Data 20 (1975) 106–109. DOI: 10.1021/je60064a006.

[97] R. Evans, G. Dal Poggetto, M. Nilsson, G. A. Morris: Improving the interpretation of small molecule diffusion coefficients, Analytical Chemistry 90 (2018) 3987–3994. DOI: 10.1021/acs.analchem.7b05032.

[98] A. Einstein: Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen, Annalen der Physik 322 (1905) 549–560. DOI: 10.1002/andp.19053220806.

[99] B. E. Poling, J. M. Prausnitz, J. P. O'Connell: The Properties of Gases and Liquids, McGraw-Hill, New York, 2001.

[100] C. A. Crutchfield, D. J. Harris: Molecular mass estimation by PFG NMR spectroscopy, J. Magn. Reson. 185 (2007) 179–182. DOI: 10.1016/j.jmr.2006.12.004.

[101] D. Li, I. Keresztes, R. Hopson, P. G. Williard: Characterization of reactive intermediates by multinuclear diffusion-ordered NMR spectroscopy (DOSY), Accounts of Chemical Research 42 (2009) 270–280. DOI: 10.1021/ar800127e.

[102] R. Neufeld, D. Stalke: Accurate molecular weight determination of small molecules via DOSY-NMR by using external calibration curves with normalized diffusion coefficients, Chemical science 6 (2015) 3354–3364. DOI: 10.1039/C5SC00670H.

[103] R. Evans: The interpretation of small molecule diffusion coefficients: Quantitative use of diffusion-ordered NMR spectroscopy, Progress in Nuclear Magnetic Resonance Spectroscopy 117 (2020) 33–69. DOI: 10.1016/j.pnmrs.2019.11.002.

[104] R. Ramprasad, R. Batra, G. Pilania, A. Mannodi-Kanakkithodi, C. Kim: Machine learning in materials informatics: recent applications and prospects, Computational Materials 3 (2017) 1–13. DOI: 10.1038/s41524-017-0056-5.

[105] K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev, A. Walsh: Machine learning for molecular and materials science, Nature 559 (2018) 547–555. DOI: 10.1038/s41586-018-0337-2.

[106] V. Venkatasubramanian: The promise of artificial intelligence in chemical engineering: Is it here, finally?, AIChE Journal 65 (2019) 466–478. DOI: 10.1002/aic.16489.

[107] A. Abbasi, R. Eslamloueyan: Determination of binary diffusion coefficients of hydrocarbon mixtures using MLP and ANFIS networks based on QSPR method, Chemometrics and Intelligent Laboratory Systems 132 (2014) 39–51. DOI: 10.1016/j.chemolab.2013.12.007.

[108] R. Beigzadeh, M. Rahimi, S. R. Shabanian: Developing a feed forward neural network multilayer model for prediction of binary diffusion coefficient in liquids, Fluid Phase Equilibria 331 (2012) 48–57. DOI: 10.1016/j.fluid.2012.06.025.

[109] F. Gharagheizi, M. Sattari: Estimation of molecular diffusivity of pure chemicals in water: a quantitative structure–property relationship study, SAR

and QSAR in Environmental Research 20 (2009) 267–285. DOI: 10.1080/10629360902949534.

[110] A. Khajeh, M. R. Rasaei: Diffusion coefficient prediction of acids in water at infinite dilution by QSPR method, Structural Chemistry 23 (2012) 399–406. DOI: 10.1007/s11224-011-9879-8.

[111] H.-J. Xue, X.-Y. Dai, J. Zhang, S. Huang, J. Chen: Deep matrix factorization models for recommender systems, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI'17, AAAI Press, Melbourne, Australia, 2017, pp. 3203–3209. DOI: 10.24963/ijcai.2017/447.

[112] M. J. Pazzani, D. Billsus: Content-based recommendation systems, in: P. Brusilovsky, A. Kobsa, W. Nejdl (Eds.), The Adaptive Web: Methods and Strategies of Web Personalization, Lecture Notes in Computer Science, Springer Berlin, Heidelberg, 2007, pp. 325–341. DOI: 10.1007/978-3-540-72079-9.

[113] R. L. Rowley, W. V. Wilding, J. L. Oscarson, Y. Yang, N. A. Zundel, T. E. Daubert, R. P. Danner: DIPPR data compilation of pure chemical properties, Design Institute for Physical Properties, AIChE, 2003. URL: https://www.aiche.org/dippr, database date: 2018, retrieved via The DIPPR Information and Data Evaluation Manager for the Design Institute for Physical Properties - Version 12.3.0 (May 2018 Public).

[114] R. E. Schapire: The strength of weak learnability, Machine Learning 5 (1990) 197–227. DOI: 10.1007/BF00116037.

[115] G. C. Cawley, N. L. C. Talbot: Efficient leave-one-out cross-validation of kernel fisher discriminant classifiers, Pattern Recognition 36 (2003) 2585–2592. DOI: 10.1016/S0031-3203(03)00136-5.

[116] D. S. Abrams, J. M. Prausnitz: Statistical thermodynamics of liquid mixtures: A new expression for the excess Gibbs energy of partly or completely miscible systems, AIChE Journal 21 (1975) 116–128. DOI: 10.1002/aic.690210115.

[117] G. Maurer, J. M. Prausnitz: On the derivation and extension of the UNIQUAC equation, Fluid Phase Equilibria 2 (1978) 91–99. DOI: 10.1016/0378-3812(78)85002-x.

[118] H. Renon, J. M. Prausnitz: Local compositions in thermodynamic excess functions for liquid mixtures, AIChE Journal 14 (1968) 135–144. DOI: 10.1002/aic.690140124.

[119] S. Skjold-Jorgensen, B. Kolbe, J. Gmehling, P. Rasmussen: Vapor-liquid equilibria by UNIFAC group contribution. Revision and extension, Industrial & Engineering Chemistry Process Design and Development 18 (1979) 714–722. DOI: 10.1021/i260072a024.

[120] J. Gmehling, P. Rasmussen, A. Fredenslund: Vapor-liquid equilibriums by UNIFAC group contribution. Revision and extension. 2, Industrial & Engineering Chemistry Process Design and Development 21 (1982) 118–127. DOI: 10.1021/i200016a021.

[121] E. A. Macedo, U. Weidlich, J. Gmehling, P. Rasmussen: Vapor-liquid equilibriums by UNIFAC group contribution. Revision and extension. 3, Industrial & Engineering Chemistry Process Design and Development 22 (1983) 676–678. DOI: 10.1021/i200023a023.

[122] D. Tiegs, P. Rasmussen, J. Gmehling, A. Fredenslund: Vapor-liquid equilibria by UNIFAC group contribution. 4. revision and extension, Industrial & Engineering Chemistry Research 26 (1987) 159–161. DOI: 10.1021/ie00061a030.

[123] H. K. Hansen, P. Rasmussen, A. Fredenslund, M. Schiller, J. Gmehling: Vapor-liquid equilibria by UNIFAC group contribution. 5. revision and extension, Industrial & Engineering Chemistry Research 30 (1991) 2352–2355. DOI: 10.1021/ie00058a017.

[124] T. Magnussen, P. Rasmussen, A. Fredenslund: UNIFAC parameter table for prediction of liquid-liquid equilibriums, Industrial & Engineering Chemistry Process Design and Development 20 (1981) 331–339. DOI: 10.1021/i200013a024.

[125] G. Wienke, J. Gmehling: Prediction of octanol-water partition coefficients, Henry coefficients and water solubilities using UNIFAC, Toxicological & Environmental Chemistry 65 (1998) 57–86. DOI: 10.1080/02772249809358557.

[126] W. Yan, M. Topphoff, C. Rose, J. Gmehling: Prediction of vapor–liquid equilibria in mixed-solvent electrolyte systems using the group contribution concept, Fluid Phase Equilibria 162 (1999) 97–113. DOI: 10.1016/S0378-3812(99)00201-0.

[127] DDBST - Dortmund Data Bank Software & Separation Technology GmbH: The unifac consortium, 2022. URL: http://www.unifac.org.

[128] A. A. Bondi: Physical Properties of Molecular Crystals Liquids, and Glasses, Wiley, 1968.

[129] A. Fredenslund: Vapor-liquid equilibria using UNIFAC: a group-contribution method, Elsevier, 2012.

[130] J. Gmehling, R. Wittig, J. Lohmann, R. Joh: A modified UNIFAC (Dortmund) model. 4. Revision and extension, Industrial & engineering chemistry research 41 (2002) 1678–1688. DOI: 10.1021/ie0108043.

[131] B. Schmid, A. Schedemann, J. Gmehling: Extension of the VTPR group contribution equation of state: Group interaction parameters for additional 192 group combinations and typical results, Industrial & Engineering Chemistry Research 53 (2014) 3393–3405. DOI: 10.1021/ie404118f.

[132] G. Takács, I. Pilászy, B. Németh, D. Tikk: Investigation of various matrix factorization methods for large recommender systems, in: 2008 IEEE International Conference on Data Mining Workshops, IEEE, 2008, pp. 553–562. DOI: 10.1145/1722149.1722155.

[133] J. G. Rittig, K. B. Hicham, A. M. Schweidtmann, M. Dahmen, A. Mitsos: Graph neural networks for temperature-dependent activity coefficient prediction of solutes in ionic liquids, arXiv preprint arXiv:2206.11776 (2022).

[134] J. M. Prausnitz, F. Anderson, T. Anderson: Computer calculations for multicomponent vapor-liquid and liquid-liquid equilibria, Prentice Hall, 1980.

[135] U. Onken, J. Rarey-Nies, J. Gmehling: The Dortmund Data Bank: A computerized system for retrieval, correlation, and prediction of thermodynamic properties of mixtures, International Journal of Thermophysics 10 (1989) 739–747. DOI: 10.1007/BF00507993.

[136] J. Gmehling: Sophisticated thermodynamic models and Dortmund Data Bank, Vakuum in Forschung und Praxis 14 (2002) 272–279. DOI: 10.1002/1522-2454(200210)14:5<272::AID-VIPR272>3.0.CO;2-H.

[137] DDBST - Dortmund Data Bank Software & Separation Technology GmbH: Dortmund Data Bank, 2022. URL: www.ddbst.com.

[138] O. Redlich, A. Kister: Algebraic representation of thermodynamic properties and the classification of solutions, Industrial & Engineering Chemistry 40 (1948) 345–348. DOI: 10.1021/ie50458a036.

[139] E. Herington: A thermodynamic test for the internal consistency of experimental data on volatility ratios, Nature 160 (1947) 610–611. DOI: 10.1038/160610b0.

[140] H. C. Van Ness, S. M. Byer, R. E. Gibbs: Vapor-liquid equilibrium: Part I. an appraisal of data reduction methods, AIChE Journal 19 (1973) 238–244. DOI: 10.1002/aic.690190206.

[141] DDBST - Dortmund Data Bank Software & Separation Technology GmbH: The unifac consortium, 2023. URL: http://www.unifac.org.

[142] DDBST - Dortmund Data Bank Software & Separation Technology GmbH: Dortmund Data Bank, 2024. URL: www.ddbst.com.

[143] Chemstations, Inc.: Chemcad v.8, 2024. URL: www.chemstations.com.

[144] Aspen Technology, Inc.: Aspen plus v14.5, 2024. URL: www.aspentech.com.

[145] N. Hayer, H. Hasse, F. Jirasek: Modified UNIFAC 2.0 – a group-contribution method completed with machine learning, Industrial & Engineering Chemistry Research (2025). DOI: 10.1021/acs.iecr.5c00077.

[146] J. Gmehling, J. Li, M. Schiller: A modified UNIFAC model. 2. Present parameter matrix and results for different thermodynamic properties, Industrial & Engineering Chemistry Research 32 (1993) 178–193. DOI: 10.1021/ie00013a024.

[147] J. Dewulf, D. Drijvers, H. van Langenhove: Measurement of Henry's law constant as function of temperature and salinity for the low temperature range, Atmospheric Environment 29 (1995) 323–331. DOI: 10.1016/1352-2310(94)00256-k.

[148] H. A. Bamford, D. L. Poster, J. E. Baker: Temperature dependence of Henry's law constants of thirteen polycyclic aromatic hydrocarbons between 4°C and 31°C, Environmental Toxicology and Chemistry 18 (1999) 1905–1912. DOI: 10.1002/etc.5620180906.

[149] J. Staudinger, P. V. Roberts: A critical compilation of Henry's law constant temperature dependence relations for organic compounds in dilute aqueous solutions, Chemosphere 44 (2001) 561–576. DOI: 10.1016/s0045-6535(00)00505-1.

[150] F. Wieland, A. Neff, A. N. Gloess, L. Poisson, S. Atlan, D. Larrain, D. Prêtre, I. Blank, C. Yeretzian: Temperature dependence of Henry's law constants: An automated, high-throughput gas stripping cell design coupled to PTR-ToF-MS, International Journal of Mass Spectrometry 387 (2015) 69–77. DOI: 10.1016/j.ijms.2015.07.015.

[151] D. M. Himmelblau: Solubilities of inert gases in water. 0 °C to near the critical point of water, Journal of Chemical & Engineering Data 5 (1960) 10–15. DOI: 10.1021/je60005a003.

[152] E. Wilhelm, R. Battino, R. J. Wilcock: Low-pressure solubility of gases in liquid water, Chemical Reviews 77 (1977) 219–262. DOI: 10.1021/cr60306a003.

[153] O. R. Quayle: The parachors of organic compounds. an interpretation and catalogue, Chemical Reviews 53 (1953) 439–589. DOI: 10.1021/cr60166a003.

[154] J. R. Partington: An Advanced Treatise on Physical Chemistry, Vol. I, Fundamental Principles: The Properties of Gases, Longmans, London, 1949.

[155] E. G. Scheibel: Correspondence. liquid diffusivities. viscosity of gases, Industrial & Engineering Chemistry 46 (1954) 2007–2008. DOI: 10.1021/ie50537a062.

[156] A. Spernol, K. Wirtz: Zur Mikroreibung in Flüssigkeiten, Zeitschrift für Naturforschung 8 (1953) 522–532. DOI: 10.1515/zna-1953-0902.

[157] D. R. Olander: The diffusivity of water in organic solvents, AIChE Journal 7 (1961) 175–176. DOI: 10.1002/aic.690070139.

Appendix

A Supporting Information for Chapter 3

A.1 Outliers of Modified UNIFAC (Dortmund)

The predictions of modified UNIFAC (Dortmund) [13] include eight extreme outliers, cf. Table A.1, which can be attributed to poorly fitted group-interaction parameters. Specifically, all of the relevant solutes contain main group 42 ("CY-CH2"), while all of the relevant solvents contain main group 18 ("PYRIDINE").

The few observed outliers would drastically increase the mean absolute error (MAE) and mean squared error (MSE) and thus lead to a false impression of the predictive performance of modified UNIFAC (Dortmund); they were therefore removed. By removing the eight listed outliers, the MAE decreases from 0.6477 to 0.3340.

Table A.1: Binary systems with available experimental $\ln \gamma_{ij}^{\infty}$ at 298.15 K where modified UNIFAC (Dortmund) predictions deviate significantly from the experimental data, likely due to inaccurate group-interaction parameters. Identifiers (DDB no.) are the original ones from the DDB [38].

Solute i		Solvent j	
DDB no.	Name	DDB no.	Name
50	Cyclohexane	19	2-Methylpyridine
50	Cyclohexane	144	Pyridine
50	Cyclohexane	433	Quinoline
51	Cyclopentane	433	Quinoline
52	Cyclohexene	144	Pyridine
159	Tetrahydrofuran	144	Pyridine
401	Ethylcyclohexane	19	2-Methylpyridine
401	Ethylcyclohexane	144	Pyridine

A.2 Results of the Hyperparameter Variations

The predictive performance of each SBM variant is shown in Fig. 3, focusing on the MAE. By displaying the MSE in a similar way, Fig. A.1 supports the choice of the final model, represented by the orange dots.

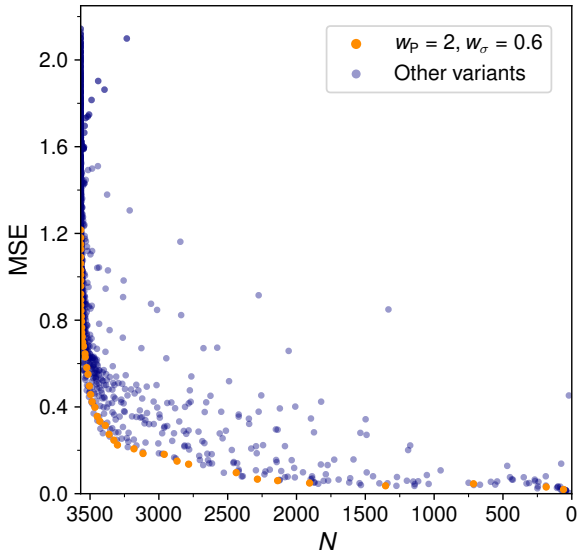


Figure A.1: Mean squared error (MSE) of the predicted $\ln \gamma_{ij}^{\infty}$ from the leave-one-out analysis over the number of predictable experimental data points N for all tested SBM variants. The results of the best-performing SBM (as specified with the weights w) are highlighted in orange.

In Fig. A.2, a detailed analysis of the results is shown, focusing on the influence of different hyperparameter choices. From this analysis, the following heuristics are derived:

- A stronger weighting of the polar regions in the σ-profiles (w_P = 2) enhances both objectives, i.e., the predictive performance and the scope, especially near the Pareto knee, cf. Fig. A.2a.
- Relying solely on surface area similarity (w_σ = 0) results in poor predictions, cf. Fig. A.2b.
- Focusing solely on charge distribution similarity (w_σ = 1) achieves comparatively good results at lower thresholds, where both the SBM scope and the MAE are generally large. However, increasing the threshold yields only small improvements in predictive accuracy, cf. Fig. A.2c. In contrast, many model variants incorporating surface area similarity ($w_\sigma < 1$) perform significantly better here.

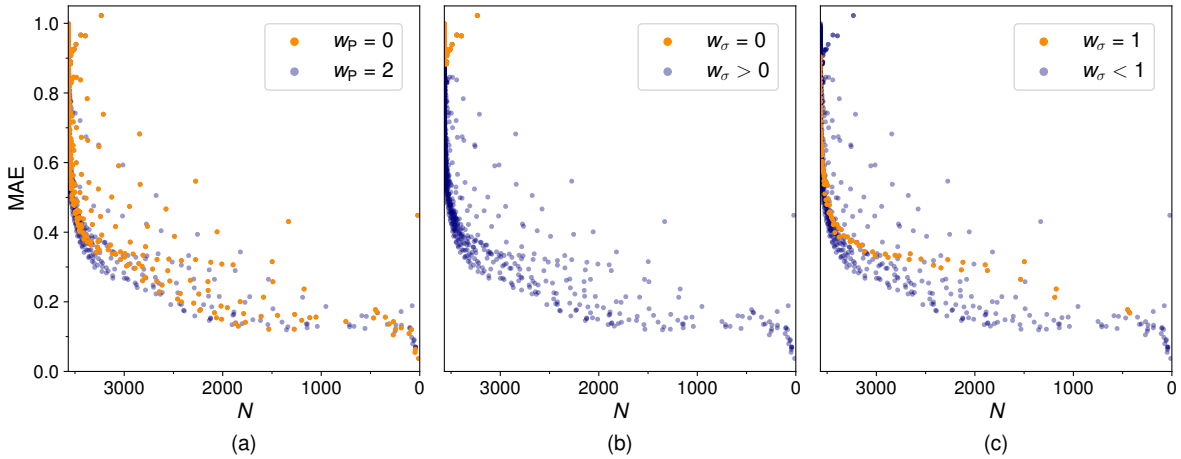


Figure A.2: Mean absolute error (MAE) of the predicted $\ln \gamma_{ij}^{\infty}$ from the leave-one-out analysis over the number of predictable experimental data points N for all tested SBM variants. In each panel, a subset of SBM variants is highlighted in orange: (a) equal weighting of the polar and non-polar regions in the σ-profiles ($w_P = 0$), (b) using only the surface area similarity ($w_\sigma = 0$), (c) using only the charge distribution similarity ($w_\sigma = 1$).

A.3 Scope of the Proposed Similarity-Based Method

The performance of the final SBM, characterized through $w_\sigma = 0.6$, and $w_P = 2$, can be influenced by varying the threshold ξ. Fig. A.3 illustrates the resulting trade-off between scope and predictive accuracy, focusing not only on the scope in terms of the number of predictable data points from the database (Fig. A.3a), but also on the number of predictable data points from all possible solute-solvent combinations (Fig. A.3b).

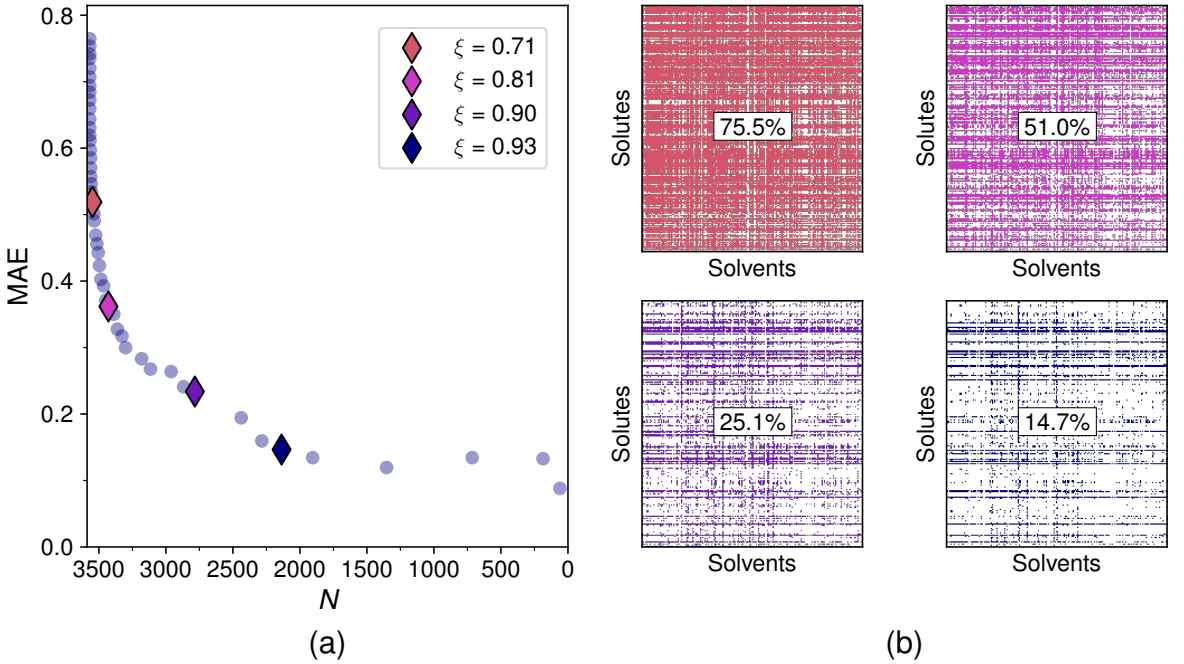


Figure A.3: Influence of the threshold ξ on the predictive performance and the scope of the SBM. (a) Mean absolute error (MAE) of the SBM for the prediction of $\ln \gamma_{ij}^{\infty}$. N represents the number of predictable experimental data points. Four distinct thresholds are highlighted. (b) Matrices representing all predictable solute-solvent combinations for the four ξ values highlighted in (a).

In general, a large ξ results in a small scope and vice versa. Thereby, the percentage of predictable data points from the experimental database is always higher than the scope concerning the entire solute-solvent matrix. For example, the selected threshold of $\xi = 0.93$ yields an SBM that can predict 59.9% of the experimental database but can only populate 14.7% of the solute-solvent matrix. The SBM is limited in extrapolating into the sparse region of the matrix, which is not surprising since it relies on experimental data points of similar mixtures. In comparison, modified UNIFAC (Dortmund) [13] is capable of predicting 83.7% of the experimentally studied mixtures and 69.9% of the entire matrix, COSMO-SAC-dsp [47] achieves 89.7% and 85.4%, respectively, whereas COSMO-SAC [46] can calculate $\ln \gamma_{ij}^{\infty}$ for all considered binary mixtures.

By reducing ξ, the scope of the SBM could be extended so that almost any solute-solvent combination can be predicted. However, this would lead to a significant loss of predictive accuracy. Therefore, the SBM with $\xi = 0.93$ continues to be used as its prediction accuracy is within the range of experimental uncertainty, and the limited scope is accepted. The unprecedented performance of this SBM makes it a precious tool in chemical process engineering, even if it is not applicable in all cases.

A.4 Case Studies of Similar Components

The calculated similarity scores S_{ij} between two components are not only the basis of the proposed SBM for the prediction of activity coefficients at infinite dilution. They also offer the possibility to identify the most similar components for a target component, exemplified in Tables A.2 and A.3. Here, the S_{ij} of the final SBM, obtained through the grid search, are used.

Table A.2: Lists of top 10 components among the solutes most similar to ethanol or n-butane.

Solutes Similar to Ethanol		Solutes Similar to n-Butane	
Solute	S_{ij}	Solute	S_{ij}
1-Propanol	0.909	Cyclopentane	0.962
2-Propanol	0.869	2-Methylpropane	0.961
Methanol	0.850	Pentane	0.930
1-Butanol	0.835	2-Methylbutane	0.929
N-Methylformamide	0.829	Propane	0.917
2-Butanol	0.808	Methylcyclopentane	0.910
1-Pentanol	0.806	Cyclohexane	0.894
tert-Butanol	0.801	3-Methylpentane	0.890
2-Methyl-1-propanol	0.776	2-Methylpentane	0.886
Cyclohexanol	0.770	2,3-Dimethylbutane	0.881

Table A.3: Lists of top 10 components among the solvents most similar to water or chlorobenzene.

Solvents Similar to Water		Solvents Similar to Chlorobenzene	
Solvent	S_{ij}	Solvent	S_{ij}
Methanol	0.716	Bromobenzene	0.977
Formamide	0.702	Iodobenzene	0.948
Ethanol	0.636	Fluorobenzene	0.921
1,2-Ethanediol	0.623	1-Chloronaphthalene	0.869
1-Propanol	0.599	1-Bromonaphthalene	0.861
2-Propanol	0.592	1,1-Dichloroethane	0.770
1,3-Propanediol	0.591	Toluene	0.746
1-Butanol	0.572	Methoxybenzene	0.743
1,4-Butanediol	0.572	Indene	0.742
1,2-Propanediol	0.569	Diiodomethane	0.740

Not surprisingly, many alkanes are similar to n-butane, and many halogenobenzenes are similar to chlorobenzene. In contrast, water has no similar components, as it is unique in being an extremely polar and rather small molecule.

B Supporting Information for Chapter 4.1

B.1 Henry's Law

Eq. (B.1) shows Henry's law for the description of gas solubilities:

$$H_{ij} \exp\left(\frac{1}{RT} \int_{p_j^s}^{p} v_{ij}^{\infty} \mathrm{d}p\right) x_i \gamma_i^* = p\, y_i \varphi_i \tag{B.1}$$

where H_{ij} represents the Henry's law constant of solute i in solvent j, R is the universal gas constant, T the temperature, p the pressure, and v_{ij}^{∞} the partial molar volume of solute i infinitely diluted in solvent j. Furthermore, x_i and y_i are the mole fractions of solute i in the liquid and vapor phase, respectively. γ_i^* represents the activity coefficient defined by a normalization of the chemical potential according to Henry's law and φ_i is the fugacity coefficient of solute i in the vapor phase.

In many cases, an incompressible liquid phase and an ideal vapor phase are assumed, which is often a valid assumption at low to moderate pressures. If, furthermore, only small concentrations of solute i in the liquid phase are considered, γ_i^* approaches unity and Eq. (B.1) simplifies to:

$$H_{ij}\, x_i = p\, y_i \tag{B.2}$$

Henry's law constants in systems with only subcritical components can in principle be calculated from activity coefficients at infinite dilution and pure component properties by applying Raoult's law, which is defined as

$$p_i^s \varphi_i^s \exp\left(\frac{1}{RT} \int_{p_i^s}^{p} v_i \mathrm{d}p\right) x_i \gamma_i = p\, y_i \varphi_i \tag{B.3}$$

where p_i^s and φ_i^s represent the pure component vapor pressure and the pure component saturated vapor fugacity coefficient of solute i. Furthermore, v_i is the molar volume of

pure component i and γ_i is the activity coefficient of solute i in the mixture. The term $\exp\left(\frac{1}{RT}\int_{p_i^s}^{p} v_i \mathrm{d}p\right)$ is called Poynting correction. Combining Eqs. (B.1) and (B.3) and assuming infinite dilution leads to

$$H_{ij} = p_i^s \, \varphi_i^s \, \exp\left(\frac{1}{RT} \int_{p_i^s}^{p} v_i \mathrm{d}p\right) \gamma_{ij}^{\infty} \tag{B.4}$$

where γ_{ij}^{∞} is the activity coefficient of solute i infinitely diluted in solvent j. Note that at infinite dilution of solute i in solvent j, the activity coefficient normalized according to Henry's law $\gamma_i^{\star}$ becomes unity by definition, as does the exponential term from Eq. (B.1), which is known as Krichevsky-Kasarnovsky correction, since the pressure p approaches the vapor pressure of the pure solvent p_j^s here.

Calculating Henry's law constants from pure component vapor pressures and activity coefficients at infinite dilution requires information on the Poynting correction and the pure component saturated vapor fugacity coefficients, which is not always available. Nevertheless, this has been tested and Henry's law constants have been calculated from activity coefficients at infinite dilution for all experimental data on H_{ij} from the DDB [81]. Large deviations between the predicted numbers and the experimental numbers have been found. Hence, including data obtained in this way in the training is expected to lead to a deterioration rather than to improvements. Furthermore, the number of additional values of H_{ij} in the present matrix that could be obtained in this way is rather small.

B.2 Database

Fig. B.1 shows the temperature distribution of all experimental data on H_{ij}.

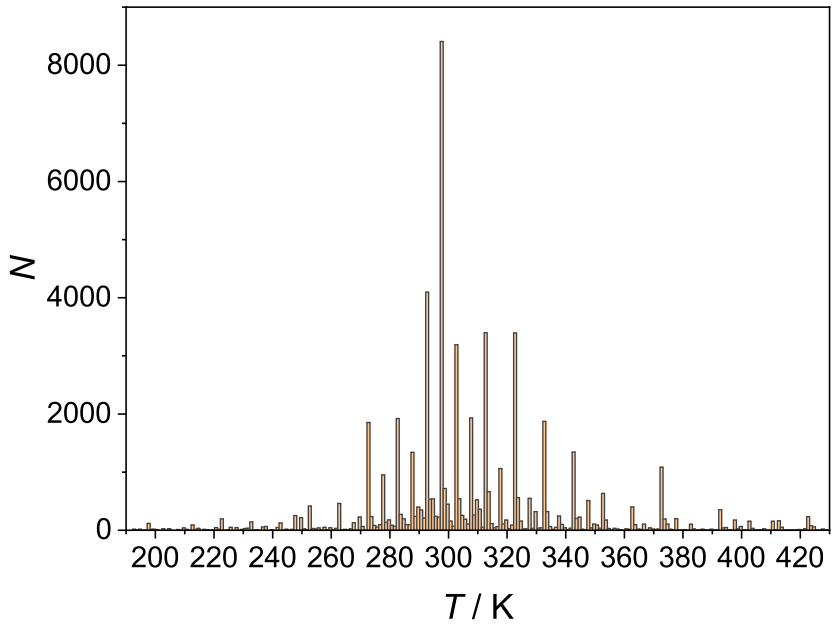


Figure B.1: Histogram of all experimental Henry's law constants from the DDB [81] over the reported temperature. Data points labeled in the DDB to be of poor quality are not shown. N is the number of data points.

As shown in Fig. B.1, the majority of data points are measured at 298.15±1 K. Therefore, this temperature range has been chosen in this chapter.

B.3 Probabilistic Model

B.3.1 Data-Driven MCM

The first MCM is purely data-driven, which means it is trained only to the sparse available experimental data on $\ln H_{ij}$; it is called MCM-data in the following.

The number of latent variables (LVs) considered per component, $K + 1$, as well as the standard deviations of the prior σ_{P} and $\sigma_{\mathrm{P,CB}}$ and of the likelihood σ_{L} are hyperparameters of the model and were set by cross-validation to: $K = 4$, $\sigma_{\mathrm{P}} = 1.0$, $\sigma_{\mathrm{P,CB}} = 10.0$, and $\sigma_{\mathrm{L}} = 0.2$.

To put it in a nutshell, the generative model of MCM-data first draws a vector $\boldsymbol{u}_i$ ($\boldsymbol{v}_j$) of length K for the LVs for each solute i (each solvent j) from a normal prior distribution with standard deviation $\sigma_{\mathrm{P}} = 1.0$ centered around zero. Additionally it draws a scalar

b_i^u (b_j^v) from a normal prior distribution with standard deviation $\sigma_{\mathrm{P,CB}}$ = 10.0. It then models the probability of each experimental $\ln H_{ij}$ as a normal distribution with standard deviation σ_{L} = 0.2 centered around $\boldsymbol{u}_i \cdot \boldsymbol{v}_j + b_i^u + b_j^v$, which is equivalent to the modeled value of $\ln H_{ij}$, cf. Eq. (9).

The Stan Code of MCM-data is shown in Fig. B.2.

```
1  data {
2      int<lower=0> I; // number of solutes
3      int<lower=0> J; // number of solvents
4      int<lower=0> K; // number of latent dimensions
5      real ln_H[I,J]; //matrix of logarithmic Henry coefficients
6      real<lower=0> lambda; // likelihood scale
7      real<lower=0> sigma_0; // prior standard deviation
8      real<lower=0> sigma_0_CompBias; // prior standard deviation of component bias
9  }
10
11 parameters {
12     vector[K] u[I]; // solute feature vectors
13     vector[K] v[J]; // solvent feature vectors
14     real u_bias[I]; // solute bias
15     real v_bias[J]; // solvent bias
16 }
17
18 model {
19     // prior: draw feature vectors for all solutes and solvents:
20     for (i in 1:I){
21         u[i] ~ normal(0,sigma_0);
22         u_bias[i] ~ normal(0,sigma_0_CompBias);
23     }
24     for (j in 1:J) {
25         v[j] ~ normal(0,sigma_0);
26         v_bias[j] ~ normal(0,sigma_0_CompBias);
27     }
28
29     // likelihood: model the probability of ln_H as a normal distribution
30     // around the dot product of the feature vectors:
31     for (i in 1:I) {
32         for (j in 1:J) {
33             if (ln_H[i,j] != -99) { // train to available data only
34                 ln_H[i,j] ~ normal(u[i]' * v[j] + u_bias[i] + v_bias[j], lambda);
35             }
36         }
37     }
38 }
```

Figure B.2: Stan code for MCM-data. Line 33 ensures that the method is only trained to the observed entries of the matrix, since all unobserved entries were set to -99 prior to the training.

B.3.2 Hybrid MCM

The second MCM is similar to the above described MCM-data and operates on the same LVs (and the same number of LVs). However, in contrast to MCM-data, this MCM additionally incorporates information from the physics-based prediction method Predictive Soave-Redlich-Kwong (PSRK) equation-of-state. This MCM is called MCM-hybrid in the following and its training consists of two steps, a *pretraining* and a *refinement* step. In the pretraining step, MCM-hybrid is not trained on experimental data but on PSRK predictions for $\ln H_{ij}$. In the second step of the training of MCM-hybrid, the refinement step, the method is trained on experimental data and an informative prior based on the posterior of the preceding pretraining step is used.

As for MCM-data, all hyperparameters of MCM-hybrid were chosen by cross-validation. However, the models appeared very robust towards variations of the hyperparameters. For the most significant hyperparameter K, a sensitivity study is given below.

The Stan Codes of MCM-hybrid are shown in Figs. B.3 and B.4. Additionally, all Stan Codes are given as separate text files in Ref. [82].

```
1  data {
2      int<lower=0> I; // number of solutes
3      int<lower=0> J; // number of solvents
4      int<lower=0> K; // number of latent dimensions
5      real ln_H[I,J]; // matrix of logarithmic Henry coefficients
6      real<lower=0> sigma_0; // prior standard deviation
7      real<lower=0> sigma_0_CompBias; // prior standard deviation of component bias
8      real<lower=0> lambda; // likelihood scale
9  }
10
11 parameters {
12     vector[K] u[I]; // solute feature vectors
13     vector[K] v[J]; // solvent feature vectors
14     real u_bias[I]; // solute bias
15     real v_bias[J]; // solvent bias
16 }
17
18 model {
19     // prior: draw feature vectors for all solutes and solvents:
20     for (i in 1:I){
21         u[i] ~ normal(0,sigma_0);
22         u_bias[i] ~ normal(0,sigma_0_CompBias);
23     }
24     for (j in 1:J) {
25         v[j] ~ normal(0,sigma_0);
26         v_bias[j] ~ normal(0,sigma_0_CompBias);
27     }
28
29     // likelihood: model the probability of ln_H as a Cauchy distribution
30     // around the dot product of the feature vectors:
31     for (i in 1:I) {
32         for (j in 1:J) {
33             if (ln_H[i,j] != -99) { // train to available data only
34                 ln_H[i,j] ~ cauchy(u[i]' * v[j] + u_bias[i] + v_bias[j], lambda);
35             }
36         }
37     }
38 }
```

Figure B.3: Stan code for the pretraining step of MCM-hybrid. Line 33 ensures that the method is only trained to the observed entries of the matrix, since all unobserved entries were set to -99 prior to the training.

```
1  data {
2      int<lower=0> I; // number of solutes
3      int<lower=0> J; // number of solvents
4      int<lower=0> K; // number of latent dimensions
5      real ln_H[I,J]; //matrix of logarithmic Henry coefficients
6      real<lower=0> lambda; // likelihood scale
7      vector<lower=0>[K] sigma_0_u[I]; // prior standard deviation (Solutes)
8      vector<lower=0>[K] sigma_0_v[J]; // prior standard deviation (Solvents)
9      real<lower=0> sigma_0_CompBias_u[I]; // prior standard deviation of component bias (Solutes)
10     real<lower=0> sigma_0_CompBias_v[J]; // prior standard deviation of component bias (Solvents)
11     vector[K] mu_0_u[I]; // prior mean (Solutes)
12     vector[K] mu_0_v[J]; // prior mean (Solvents)
13     real mu_0_CompBias_u[I]; // prior mean of component bias (Solutes)
14     real mu_0_CompBias_v[J]; // prior mean of component bias (Solvents)
15 }
16
17 parameters {
18     vector[K] u[I]; // solute feature vectors
19     vector[K] v[J]; // solvent feature vectors
20     real u_bias[I]; // solute bias
21     real v_bias[J]; // solvent bias
22 }
23
24 model {
25     // prior: draw feature vectors for all solutes and solvents:
26     for (i in 1:I){
27         u[i] ~ normal(mu_0_u[i],sigma_0_u[i]);
28         u_bias[i] ~ normal(mu_0_CompBias_u[i],sigma_0_CompBias_u[i]);
29     }
30     for (j in 1:J) {
31         v[j] ~ normal(mu_0_v[j],sigma_0_v[j]);
32         v_bias[j] ~ normal(mu_0_CompBias_v[j],sigma_0_CompBias_v[j]);
33     }
34
35     // likelihood: model the probability of ln_H as a normal distribution
36     // around the dot product of the feature vectors:
37     for (i in 1:I) {
38         for (j in 1:J) {
39             if (ln_H[i,j] != -99) { // train to available data only
40                 ln_H[i,j] ~ normal(u[i]' * v[j] + u_bias[i] + v_bias[j], lambda);
41             }
42         }
43     }
44 }
```

Figure B.4: Stan code for the refinement step of MCM-hybrid. Line 39 ensures that the method is only trained to the observed entries of the matrix, since all unobserved entries were set to -99 prior to the training.

B.4 Calculation of Model Predictions

After the training of the matrix completion methods (MCMs), the resulting posterior probability distributions of all latent variables (LV) were used to calculate probability distributions for the predictions of Henry's law constants $\ln H_{ij}$ for all combinations of the studied solutes i and solvents j by sampling from the posterior. The mean for each $\ln H_{ij}$ was considered as predicted data point and compared to experimental data and PSRK predictions if the respective experimental data point (PSRK prediction) was available.

All predictions for $\ln H_{ij}$ discussed in this chapter were thereby obtained after training the MCMs to all available experimental data except for the data on the respective system $i - j$ in a so-called leave-one-out analysis. Hence, the respective $\ln H_{ij}$ was always excluded from the training data, which ensures that the method cannot cheat by being trained to the $\ln H_{ij}$ to be predicted.

B.5 Additional Results

B.5.1 PSRK Outliers

The predictions with the Predictive Soave-Redlich-Kwong (PSRK) equation-of-state include several extreme outliers, which strongly deviate from the experimental data as shown in Fig. B.5. Most of these outliers correspond to systems of the solute hydrochloric acid (HCl) in alcoholic solvents.

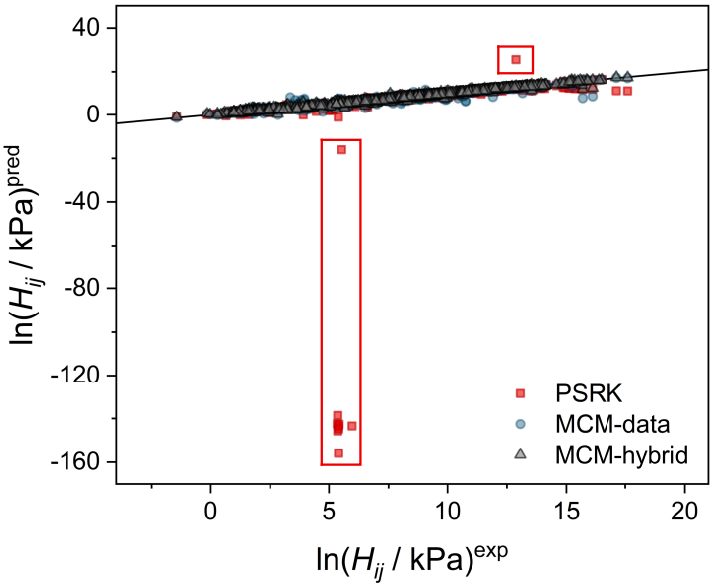


Figure B.5: Parity plot of the predictions (pred) for $\ln H_{ij}$ with PSRK, MCM-data, and MCM-hybrid over the experimental data (exp) from the DDB [81]. The worst 11 outliers of PSRK are marked.

In contrast to these findings, all proposed matrix completion methods (MCM) show a robust performance on all available data and do not show extreme outliers. Even the hybrid method MCM-hybrid, which combines PSRK with a data-driven MCM, does not suffer from the extreme PSRK outliers if they are used during the training (in the pretraining step).

B.5.2 Data-Driven MCM without Component Bias

In a simpler variant of MCM-data, referred to as MCM-0 in the following, no solute and solvent biases were considered, i.e., $b_i^u = b_j^v = 0 \ \forall \ i, j$, and, hence, $\ln H_{ij}$ was modeled as the product $\boldsymbol{u}_i \cdot \boldsymbol{v}_j$ alone, cf. Eq. (9). In analogy to MCM-data, $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$ were trained only to the available experimental data for $\ln H_{ij}$ from the DDB, i.e., MCM-0 is entirely data-driven. Except for neglecting b_i^u and b_j^v, MCM-0 is identical to MCM-data,

including the choice of the hyperparameters. Hence, also MCM-0 constitutes a purely data-driven method relying on a rather uninformative prior. MCM-0 is also similar to the MCM from Ref. [9] for the prediction of activity coefficients at infinite dilution.

The performance of MCM-0 in terms of mean absolute error (MAE) and mean squared error (MSE) is depicted in Fig. B.6 and compared to the respective scores of PSRK, MCM-data, and MCM-hybrid.

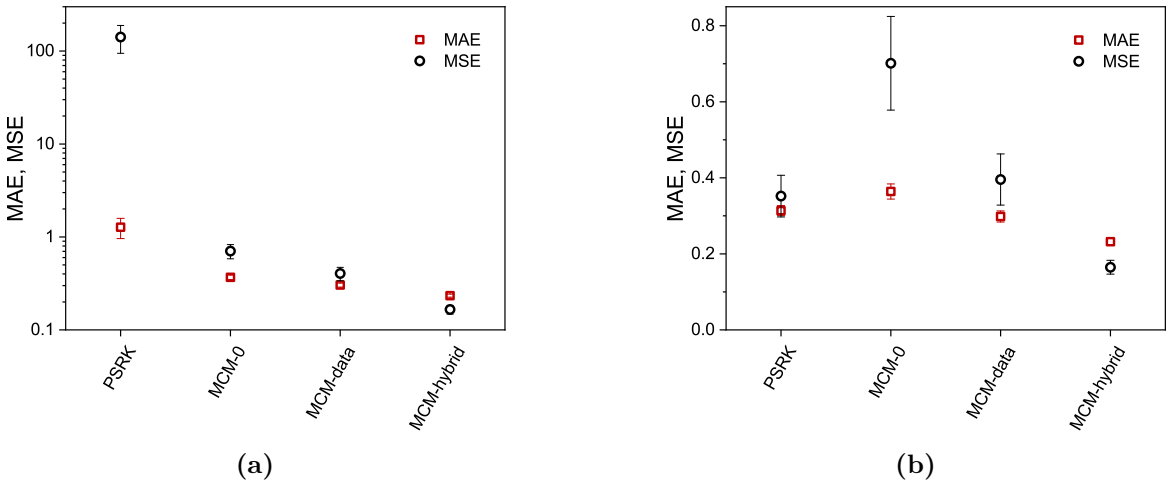


Figure B.6: Mean absolute error (MAE) and mean squared error (MSE) of PSRK and the MCMs developed in this chapter for the prediction of $\ln H_{ij}$ for binary systems at 298 K. (a) Considering the full data set (1,438 data points). (b) Without considering the worst 11 outliers of PSRK, cf. Fig. B.5.

The performance of MCM-0 is worse than that of MCM-data and MCM-hybrid and, if the extreme PSRK outliers are omitted, cf. Fig. B.6b, also worse than that of PSRK. The results demonstrate that the consideration of component biases is beneficial in MCMs for the prediction of $\ln H_{ij}$. This agrees well with the expectations, in particular if the solute-specific *general* solubility in different solvents is considered, cf. Fig. 7.

In Figs. B.7 and B.8 the performance of MCM-0, MCM-data, and MCM-hybrid for the prediction of *all* available experimental data on $\ln H_{ij}$, irrespective whether they can be predicted with PSRK, is compared. Again, the best performance is obtained with MCM-hybrid, the worst with MCM-0.

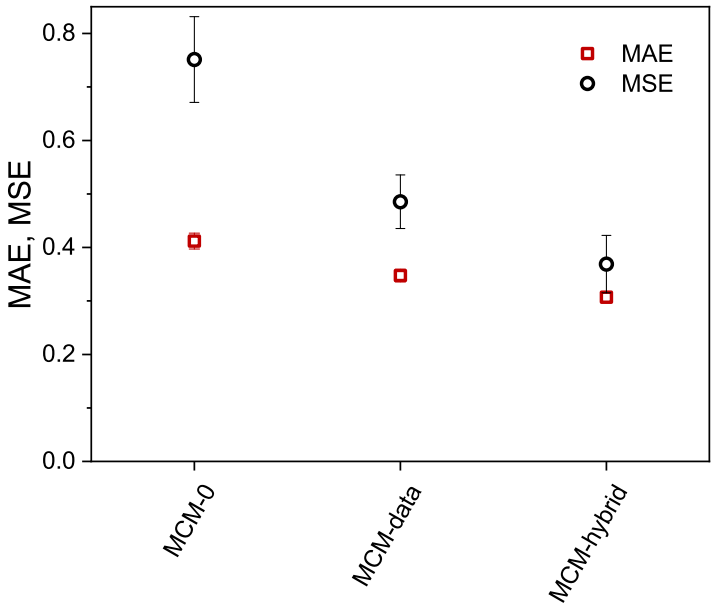
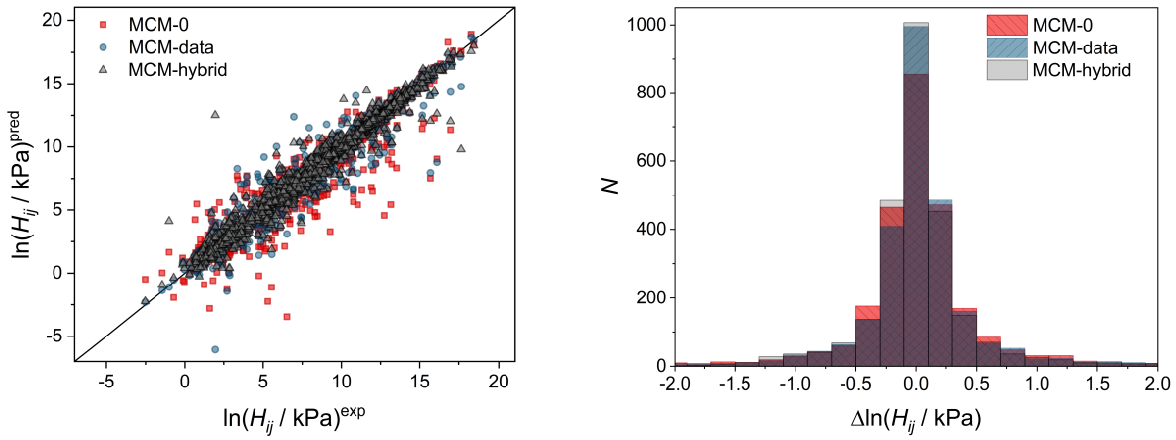


Figure B.7: Mean absolute error (MAE) and mean squared error (MSE) of the three MCMs developed in this chapter for the prediction of Henry's law constants in binary systems at 298 K considering *all* 2,661 experimental data points from the DDB.



(a) **(b)**

Figure B.8: Comparison of the predictions (pred) for $\ln H_{ij}$ with MCM-0, MCM-data, and MCM-hybrid considering *all* 2,661 experimental data points from the DDB. (a) Parity plot of predictions over experimental data (exp) from the DDB. (b) Histogram of the deviations of the predictions from the experimental data. N is the number of binary systems. The shown interval in the histogram contains 96.2 % (MCM-0), 97.6 % (MCM-data), and 98.5 % (MCM-hybrid) of the considered data points, respectively.

B.5.3 Special Case: Solute and Solvent are Identical

The studied data set includes 29 components that are considered as both solutes i and solvents j. For $i = j$, the respective entries of the matrix pertain to pure components for which, in principle, also predictions can be obtained by the MCMs. In this case and assuming an ideal vapor phase ($\varphi_i^s = 1$), the Henry's law constant H_{ii} corresponds to the vapor pressure of the pure component i. In Fig. B.9, the predictions for these components with MCM-hybrid are studied and compared to the vapor pressures obtained with the Antoine equation [72], if the respecite parameters were available in the DDB.

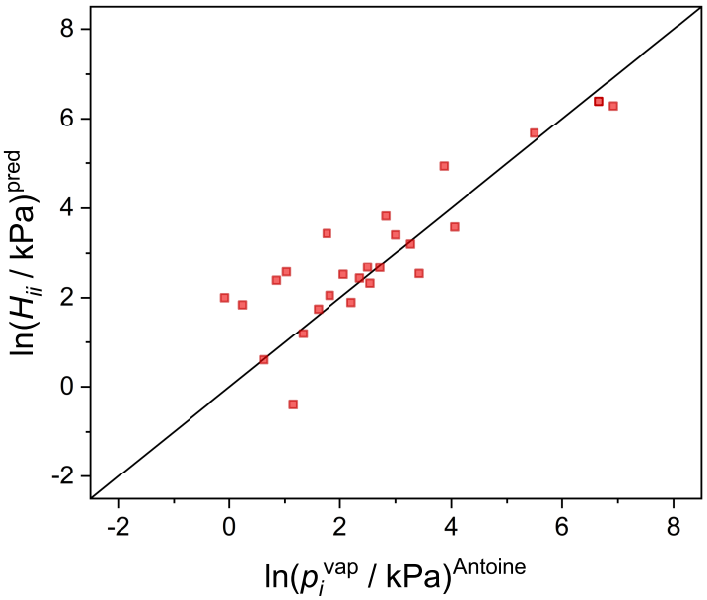


Figure B.9: Parity plot of predictions of $\ln H_{ij}$ with $i = j$ obtained with MCM-hybrid (pred) over the respective pure component vapor pressures at 298 K calculated with the Antoine equation and Antoine parameters from the DDB (Antoine).

To obtain the predictions in Fig. B.9, MCM-hybrid was trained to all experimental data on $\ln H_{ij}$ (for true mixtures, i.e., $i \neq j$, cf. reported parameters in Ref. [82]). With the Antoine parameters available in the DDB [81], 25 of the 29 respective pure components could be calculated; hence, only the predictions for those components are shown in Fig. B.9. It is interesting that the general trend of the vapor pressure of the pure components is well reproduced and predicted by the MCM after the training on mixture data only.

B.5.4 Influence of the Number of Latent Variables

K denotes the number of latent variables (LV) that are, in addition to one component bias for each solute and solvent, considered by the MCMs for each component. K is a hyperparameter of the MCMs and was chosen by cross-validation. However, in Fig. B.10, it is shown show that the influence of K over a broad range on the predictive performance of MCM-0 is rather small, i.e., that the MCM is quite robust towards variations of K. MCM-0 is by nature the most sensitive of the studied MCMs with regard to variations of K (MCM-data and MCM-hybrid both consider one additional LV for each component), which is why it was chosen for this sensitivity study here. Furthermore, similar behavior was also found for the other hyperparameters.

For the results shown in Fig. B.10, K was varied from 1 to 15, and for each case, the corresponding mean absolute error (MAE) and mean squared error (MSE) for the predictions obtained by a leave-one-out analysis are shown.

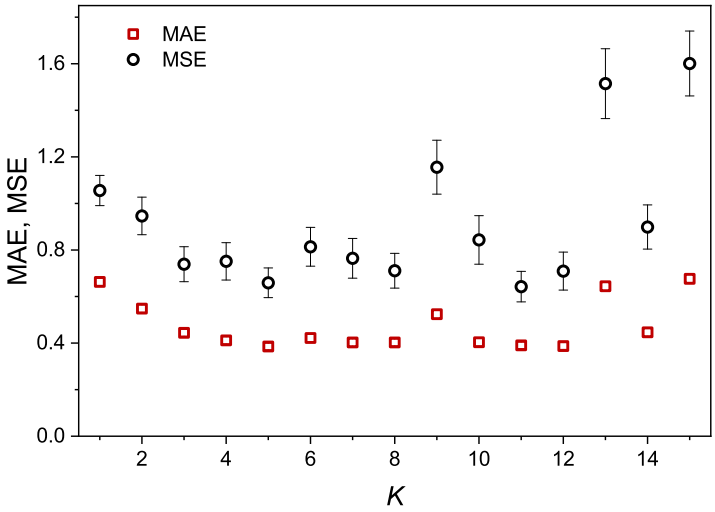
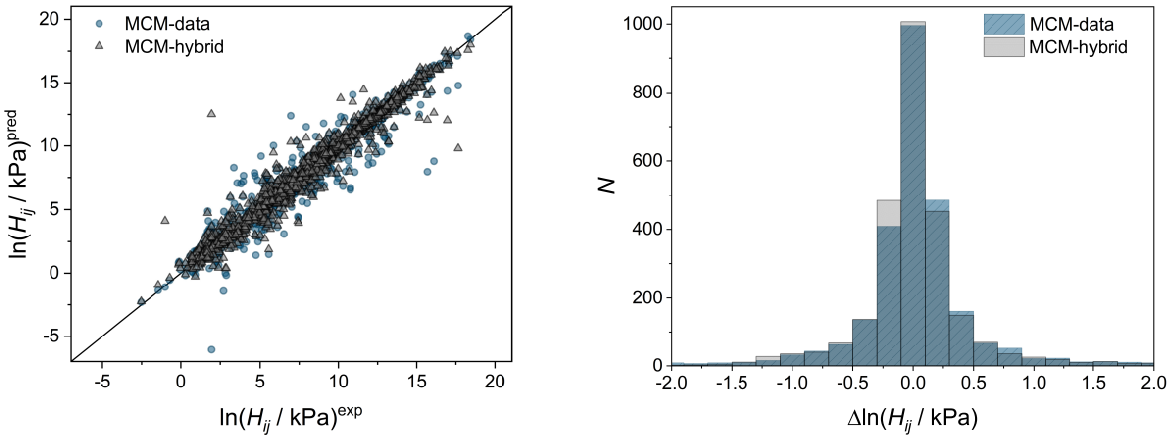


Figure B.10: Influence of the number of LVs K of MCM-0 on the predictive performance (mean absolute error (MAE) and mean squared error (MSE)) of the method.

The scores displayed in Fig. B.10 indicate that $K < 3$ leads to underfitting, while $K > 12$ leads to overfitting. Hence, the number of LVs that are considered for MCM-0 can be chosen over a broad range without notably impairing its predictive performance. In this chapter, $K = 4$ was chosen since good results were achieved with this number without unnecessarily complicating the model.

B.5.5 Predictive Performance Based on All Experimental Data

Unlike the proposed MCMs, PSRK is limited to those components and systems for which the method has been parameterized. By using the latest published parameterization [76], PSRK can predict $\ln H_{ij}$ for only 1,438 binary systems for which experimental data are available in the DDB. In contrast, the MCMs developed in this chapter allow the prediction of H_{ij} for all possible binary systems of the considered solutes and solvents. This allows the evaluation of the predictive performance of the MCMs based on all 2,661 available experimental data points, which is shown in Fig. B.11 for MCM-data and MCM-hybrid similarly to Fig. 11.



(a) (b)

Figure B.11: Comparison of the predictions (pred) for $\ln H_{ij}$ with MCM-data and MCM-hybrid considering *all* 2,661 experimental data points from the DDB. (a) Parity plot of predictions over experimental data (exp) from the DDB. (b) Histogram of the deviations of the predictions from the experimental data. N is the number of binary systems. The shown interval in the histogram contains 97.6 % (MCM-data) and 98.5 % (MCM-hybrid) of the considered data points, respectively.

B.5.6 Prediction Uncertainties

In Fig. B.12, the results of MCM-hybrid for all considered experimental data points from the DDB are shown with error bars denoting the standard deviations of the predictions, which are considered here as a measure of the model uncertainty.

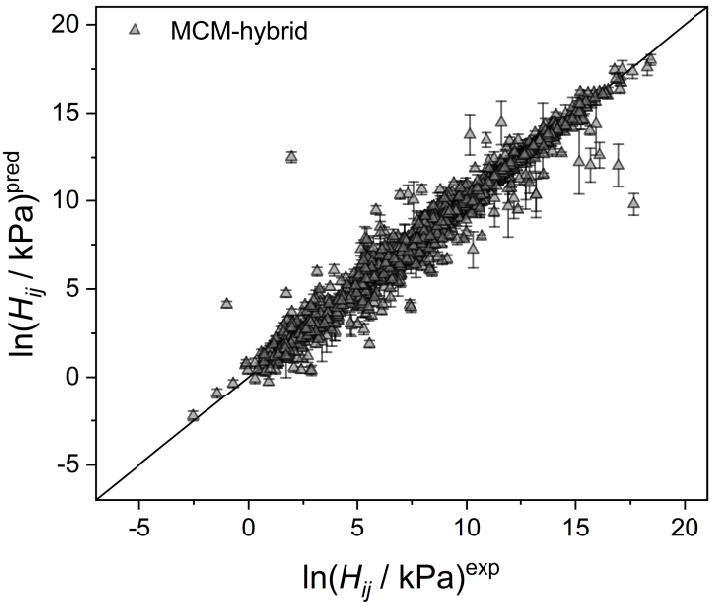


Figure B.12: Parity plot of predictions (pred) for $\ln H_{ij}$ with MCM-hybrid over *all* 2,661 experimental data points (exp) from the DDB. The error bars correspond to the calculated standard deviations.

The standard deviation of a predicted $\ln H_{ij}$ depends strongly on the number of experimental data points available for solute i and solvent j, N_i and N_j, respectively. If, as an example, only the predictions for those systems, of which at least one component is only two times represented in the data set, i.e., $N_i = 2$ or $N_j = 2$ or both, a mean standard deviation of 0.49 is obtained. By contrast, if the predictions for all other systems are considered, i.e., for $N_i \geq 3$ and $N_j \geq 3$, the mean standard deviation is 0.16.

B.5.7 Analysis of the Latent Variables

In Figs. B.13 and B.14, the LVs (except for the component biases), i.e., $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$, for all studied solutes and solvents as inferred by MCM-hybrid are shown.

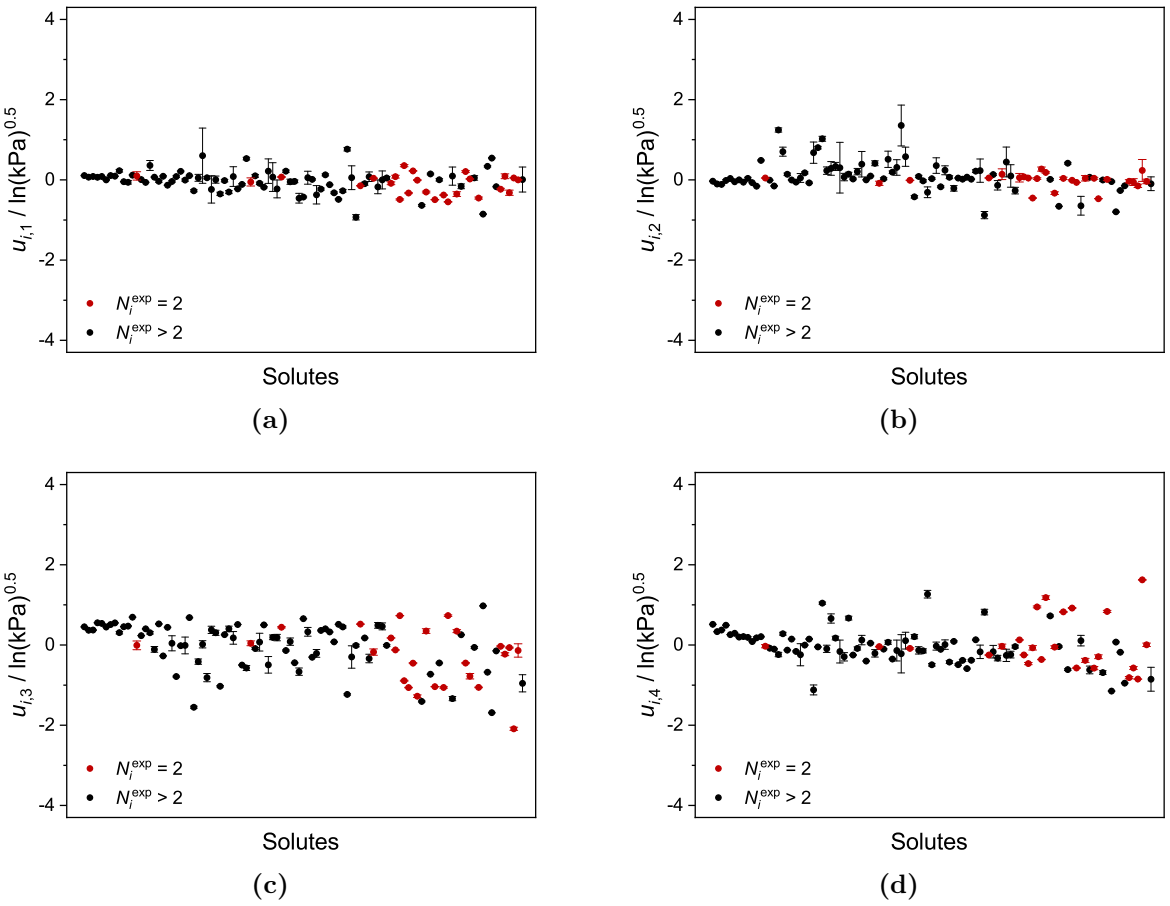


Figure B.13: LVs of all solutes, sorted in ascending order according to the critical temperature, as inferred by MCM-hybrid. Means (symbols) and standard deviations (error bars) were calculated from the results of the leave-one-out runs assuming normal distributions for the predictions. Solutes for which only data for two different systems are available in the data set are marked red. (a) $u_{i,1}$. (b) $u_{i,2}$. (c) $u_{i,3}$. (d) $u_{i,4}$.

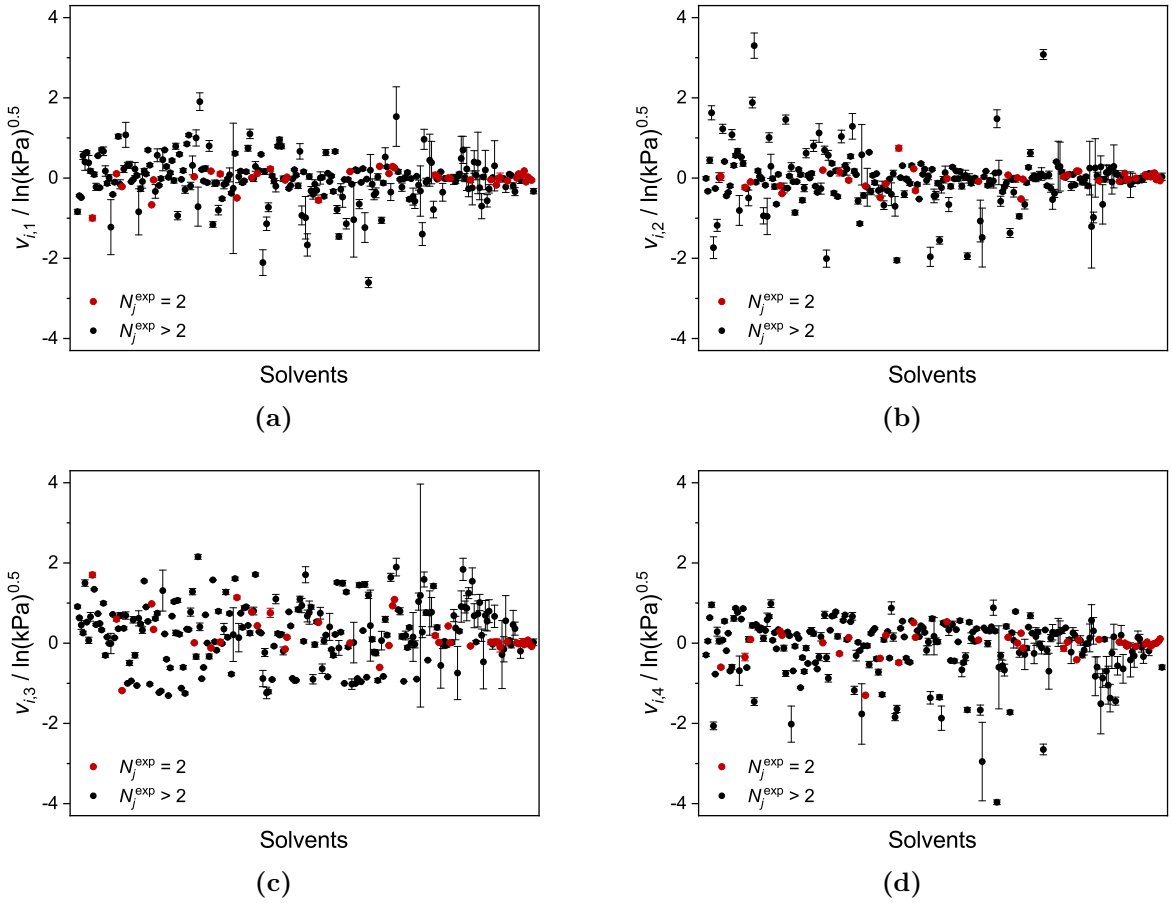


Figure B.14: LVs of all solvents, sorted in ascending order according to the DDB number, as inferred by MCM-hybrid. Means (symbols) and standard deviations (error bars) were calculated from the results of the leave-one-out runs assuming normal distributions for the predictions. Solvents for which only data for two different systems are available in the data set are marked red. (a) $v_{i,1}$. (b) $v_{i,2}$. (c) $v_{i,3}$. (d) $v_{i,4}$.

Fig. 13 shows that the variation in the component bias of the solutes is substantially higher compared to the component bias of the solvents. The remaining LVs behave in the opposite way: there is a more substantial variation in $\boldsymbol{v}_j$, cf. Fig. B.14, than in $\boldsymbol{u}_i$, cf. Fig. B.13. Furthermore, no clear correlation between the number of different binary systems for each component (N_i, N_j) and the standard deviation can be found, but Figs. B.13 and B.14 show that N_i = 2 and N_j = 2 often lead to a rather low standard deviation.

B.6 Parameter Set of the "Final" Model

In a separate excel sheet, Ref. [82] provides the final parameters of MCM-hybrid that has been trained to *all* 2,661 experimental data points for $\ln H_{ij}$ (without leave-one-out analysis). These parameters, which are posterior means (μ) of all latent variables, allow the prediction of $\ln H_{ij}$ for *any* binary combination of the solutes and solvents included in the training set by Eq. (B.5):

$$\ln H_{ij} = \boldsymbol{\mu}_{\boldsymbol{u}_i} \cdot \boldsymbol{\mu}_{\boldsymbol{v}_j} + \mu_{b_i^u} + \mu_{b_j^v} \tag{B.5}$$

The excel sheet also contains the predicted mean and standard deviation of $\ln H_{ij}$ for all possible binary combinations of the studied solutes i and solvents j obtained by sampling from the posterior.

B.7 Overview of the Studied Solutes and Solvents

Tables B.1 and B.2 contain the names, chemical formulas, and CAS numbers of all considered solutes and solvents.

Table B.1: Considered solutes (101). All H_{ij} data are extracted from the DDB [81].

Component name	**Chemical formula**	**CAS number**
Acetonitrile	C2H3N	75-05-8
Acetone	C3H6O	67-64-1
Ethanol	C2H6O	64-17-5
2-Butanol	C4H10O	78-92-2
Ethylbenzene	C8H10	100-41-4
Chlorobenzene	C6H5Cl	108-90-7
Benzene	C6H6	71-43-2
1-Butanol	C4H10O	71-36-3
2-Butanone	C4H8O	78-93-3
n-Butane	C4H10	106-97-8
Chloroform	CHCl3	67-66-3
1,2-Dichloroethane	C2H4Cl2	107-06-2

Table B.1 continued.

Component name	**Chemical formula**	**CAS number**
Dichloromethane	CH2Cl2	75-09-2
1,4-Dioxane	C4H8O2	123-91-1
Hydrogen bromide	HBr	10035-10-6
Hexane	C6H14	110-54-3
Hydrogen fluoride	HF	7664-39-3
Heptane	C7H16	142-82-5
2-Propanol	C3H8O	67-63-0
Methanol	CH4O	67-56-1
Nitromethane	CH3NO2	75-52-5
Octane	C8H18	111-65-9
2-Pentanone	C5H10O	107-87-9
Phosgene	CCl2O	75-44-5
1-Propanol	C3H8O	71-23-8
Hydrogen chloride	HCl	7647-01-0
Carbon disulfide	CS2	75-15-0
Tetrachloromethane	CCl4	56-23-5
Toluene	C7H8	108-88-3
Triethylamine	C6H15N	121-44-8
Water	H2O	7732-18-5
Ammonia	H3N	7664-41-7
Propane	C3H8	74-98-6
Chlorodifluoromethane [R22]	CHClF2	75-45-6
Dichlorodifluoromethane [R12]	CCl2F2	75-71-8
Perfluoropropylene	C3F6	116-15-4
Perfluorocyclobutane [RC318]	C4F8	115-25-3
2-Methylpropane	C4H10	75-28-5
Chloroethane	C2H5Cl	75-00-3
Furan	C4H4O	110-00-9
1,3-Butadiene	C4H6	106-99-0

Table B.1 continued.

Component name	**Chemical formula**	**CAS number**
1-Butene	C4H8	106-98-9
Chlorotrifluoromethane [R13]	CClF3	75-72-9
trans-2-Butene	C4H8	624-64-6
Isobutylene	C4H8	115-11-7
1,3,5-Trimethylbenzene	C9H12	108-67-8
Trimethylamine	C3H9N	75-50-3
Chloroperfluoroethane [R115]	C2ClF5	76-15-3
1,2-Dichlorotetrafluoroethane [R114]	C2Cl2F4	76-14-2
Dimethyl ether	C2H6O	115-10-6
Methyl chloride	CH3Cl	74-87-3
1,1-Difluoroethane [R152a]	C2H4F2	75-37-6
Ethanethiol	C2H6S	75-08-1
1-Chloro-1,2,2,2-tetrafluoroethane [R124]	C2HClF4	2837-89-0
Tetrafluoromethane [R14]	CF4	75-73-0
Carbon dioxide	CO2	124-38-9
Methane	CH4	74-82-8
Oxygen	O2	7782-44-7
Ethylene	C2H4	74-85-1
Ethane	C2H6	74-84-0
Propylene	C3H6	115-07-1
Nitrogen	N2	7727-37-9
Carbon monoxide	CO	630-08-0
Argon	Ar	7440-37-1
Chlorine	Cl2	7782-50-5
Krypton	Kr	7439-90-9
Dinitrogen monoxide	N2O	10024-97-2
Xenon	Xe	7440-63-3
Hydrogen	H2	1333-74-0
Ethyne	C2H2	74-86-2

Table B.1 continued.

Component name	**Chemical formula**	**CAS number**
Hydrogen sulfide	H2S	7783-06-4
Fluoroform [R23]	CHF3	75-46-7
Difluoromethane [R32]	CH2F2	75-10-5
Trichlorofluoromethane [R11]	CCl3F	75-69-4
Hexafluoroethane [R116]	C2F6	76-16-4
Pentafluoroethane [R125]	C2HF5	354-33-6
Helium	He	7440-59-7
Neon	Ne	7440-01-9
Sulfur hexafluoride	F6S	2551-62-4
Sulfur dioxide	O2S	7446-09-5
Perfluoropropane [R218]	C3F8	76-19-7
Deuterium	D2	7782-39-0
Tetrafluoroethylene	C2F4	116-14-3
Nitrogen oxide	NO	10102-43-9
Carbonyl sulfide	COS	463-58-1
Radon	Rn	10043-92-2
1-Chloro-1,1-difluoroethane [R142b]	C2H3ClF2	75-68-3
Diacetylene	C4H2	460-12-8
Methyl fluoride [R41]	CH3F	593-53-3
Cyclopropane	C3H6	75-19-4
Cyclobutane	C4H8	287-23-0
Propyne	C3H4	74-99-7
Sulfur trioxide	O3S	7446-11-9
Ozone	O3	10028-15-6
1,1,1,2-Tetrafluoroethane [R134a]	C2H2F4	811-97-2
1,1-Dichloro-1-fluoroethane [R141b]	C2H3Cl2F	1717-00-6
1,1,1,2,3,3,3-Heptafluoropropane [R227ea]	C3HF7	431-89-0
Perchlorylfluoride	ClFO3	7616-94-6
1-Chloro-2,2,2-trifluoroethane [R133a]	C2H2ClF3	75-88-7

Table B.1 continued.

Component name	Chemical formula	CAS number
1-Chloro-2,2-difluoroethylene (R1122)	C2HClF2	359-10-4
Chloroamine	H2ClN	10599-90-3

Table B.2: Considered solvents (247). All H_{ij} data are extracted from the DDB [81].

Component name	Chemical formula	CAS number
Acetonitrile	C2H3N	75-05-8
Acetone	C3H6O	67-64-1
1,2-Dibromoethane	C2H4Br2	106-93-4
Ethyl bromide	C2H5Br	74-96-4
1,2-Ethanediol	C2H6O2	107-21-1
Ethanol	C2H6O	64-17-5
Diethyl ether	C4H10O	60-29-7
Ethylene oxide	C2H4O	75-21-8
Formic acid	CH2O2	64-18-6
Aniline	C6H7N	62-53-3
Methoxybenzene	C7H8O	100-66-3
2-Methylpyridine	C6H7N	109-06-8
Ethyl acetate	C4H8O2	141-78-6
2-Butanol	C4H10O	78-92-2
Benzyl alcohol	C7H8O	100-51-6
Ethylbenzene	C8H10	100-41-4
Bromobenzene	C6H5Br	108-86-1
Chlorobenzene	C6H5Cl	108-90-7
Benzonitrile	C7H5N	100-47-0
Nitrobenzene	C6H5NO2	98-95-3
Benzene	C6H6	71-43-2
2-Butoxyethanol	C6H14O2	111-76-2
1-Butanol	C4H10O	71-36-3

Table B.2 continued.

Component name	**Chemical formula**	**CAS number**
2-Butanone	C4H8O	78-93-3
n-Butane	C4H10	106-97-8
Chloroform	CHCl3	67-66-3
3-Methylphenol	C7H8O	108-39-4
Cyclohexane	C6H12	110-82-7
Cyclohexene	C6H10	110-83-8
Methylcyclohexane	C7H14	108-87-2
2-Methylcyclohexanone	C7H12O	583-60-8
Dibutyl ether	C8H18O	142-96-1
Decane	C10H22	124-18-5
N,N-Dimethylaniline	C8H11N	121-69-7
1,2-Dichloroethane	C2H4Cl2	107-06-2
Dichloromethane	CH2Cl2	75-09-2
N,N-Dimethylformamide (DMF)	C3H7NO	68-12-2
1,4-Dioxane	C4H8O2	123-91-1
2,6-Dimethylpyridine	C7H9N	108-48-5
Dodecane	C12H26	112-40-3
Benzaldehyde	C7H6O	100-52-7
Butyl acetate	C6H12O2	123-86-4
Methyl acetate	C3H6O2	79-20-9
Acetic acid	C2H4O2	64-19-7
Hexane	C6H14	110-54-3
Heptane	C7H16	142-82-5
Hydrazine	H4N2	302-01-2
2-Propanol	C3H8O	67-63-0
Diisopropyl ether	C6H14O	108-20-3
2,2,4-Trimethylpentane	C8H18	540-84-1
1-Hexene	C6H12	592-41-6
Methanol	CH4O	67-56-1

Table B.2 continued.

Component name	**Chemical formula**	**CAS number**
2-Methoxyethanol	C3H8O2	109-86-4
2-Methyl-1-propanol	C4H10O	78-83-1
Nitromethane	CH3NO2	75-52-5
1-Nonanol	C9H20O	143-08-8
Octane	C8H18	111-65-9
1-Octene	C8H16	111-66-0
Pentane	C5H12	109-66-0
1-Pentanol	C5H12O	71-41-0
1-Propanol	C3H8O	71-23-8
Propionic acid	C3H6O2	79-09-4
Pyridine	C5H5N	110-86-1
Nitric acid	HNO3	7697-37-2
Carbon disulfide	CS2	75-15-0
Dimethyl sulfoxide	C2H6OS	67-68-5
tert-Butanol	C4H10O	75-65-0
Tetradecane	C14H30	629-59-4
Tetrachloromethane	CCl4	56-23-5
Tetrahydrofuran	C4H8O	109-99-9
Toluene	C7H8	108-88-3
Triethylamine	C6H15N	121-44-8
1',1',1'-Trifluorotoluene	C7H5F3	98-08-8
Water	H2O	7732-18-5
m-Xylene	C8H10	108-38-3
p-Xylene	C8H10	106-42-3
Nitroethane	C2H5NO2	79-24-3
N,N-Diethylaniline	C10H15N	91-66-7
Fluorobenzene	C6H5F	462-06-6
1,1,2,2-Tetrachloroethane	C2H2Cl4	79-34-5
Ammonia	H3N	7664-41-7

Table B.2 continued.

Component name	**Chemical formula**	**CAS number**
1,1-Dimethylhydrazine	C2H8N2	57-14-7
1,2-Dimethoxyethane	C4H10O2	110-71-4
1,1,2-Trichloro-1,2,2-trifluoroethane [R113]	C2Cl3F3	76-13-1
N-Methylformamide	C2H5NO	123-39-7
N,N-Dimethylacetamide	C4H9NO	127-19-5
Glycerol	C3H8O3	56-81-5
Acetic anhydride	C4H6O3	108-24-7
Butyric acid	C4H8O2	107-92-6
Propyl acetate	C5H10O2	109-60-4
Cyclopentanone	C5H8O	120-92-3
Cyclohexanone	C6H10O	108-94-1
Cyclohexanol	C6H12O	108-93-0
3-Methyl-1-butanol	C5H12O	123-51-3
2-Ethoxyethanol	C4H10O2	110-80-5
1,2-Propanediol	C3H8O2	57-55-6
N-Methyl-2-pyrrolidone	C5H9NO	872-50-4
3-Pentanone	C5H10O	96-22-0
2-Methyltetrahydrofuran	C5H10O	96-47-9
1-Hexanol	C6H14O	111-27-3
Hexafluorobenzene	C6F6	392-56-3
Cyclohexylamine	C6H13N	108-91-8
Perfluoro-n-heptane	C7F16	335-57-9
Perfluorotributylamine	C12F27N	311-89-7
2-Methylpropanoic acid	C4H8O2	79-31-2
cis-Decahydronaphthalene	C10H18	493-01-6
o-Xylene	C8H10	95-47-6
N-Methylaniline	C7H9N	100-61-8
1-Heptanol	C7H16O	111-70-6
1-Octanol	C8H18O	111-87-5

Table B.2 continued.

Component name	**Chemical formula**	**CAS number**
1-Decanol	C10H22O	112-30-1
Isopropylbenzene	C9H12	98-82-8
Propylbenzene	C9H12	103-65-1
Ethyl butyrate	C6H12O2	105-54-4
Acetophenone	C8H8O	98-86-2
Isobutyl acetate	C6H12O2	110-19-0
Cyclooctane	C8H16	292-64-8
Tridecane	C13H28	629-50-5
Nonane	C9H20	111-84-2
Dipropylene glycol	C6H14O3	25265-71-8
Quinoline	C9H7N	91-22-5
Triethylene glycol	C6H14O4	112-27-6
Chlorocyclohexane	C6H11Cl	542-18-7
Diethylene glycol monomethyl ether	C5H12O3	111-77-3
Carbonic acid dimethyl ester	C3H6O3	616-38-6
Diethylene glycol ethyl ether	C6H14O3	111-90-0
Diethylene glycol	C4H10O3	111-46-6
Perfluorohexane	C6F14	355-42-0
1,3,5-Trimethylbenzene	C9H12	108-67-8
o-Nitrotoluene	C7H7NO2	88-72-2
m-Nitrotoluene	C7H7NO2	99-08-1
Diethylene glycol diethyl ether	C8H18O3	112-36-7
Octamethylcyclotetrasiloxane	C8H24O4Si4	556-67-2
Diphenyl ether	C12H10O	101-84-8
1-Undecanol	C11H24O	112-42-5
Hexadecane	C16H34	544-76-3
Phthalic acid dibutyl ester	C16H22O4	84-74-2
1,2,4-Trimethylbenzene	C9H12	95-63-6
O-Deuteromethanol	CH3DO	1455-13-6

Table B.2 continued.

Component name	**Chemical formula**	**CAS number**
1-Dodecanol	C12H26O	112-53-8
Sulfolane	C4H8O2S	126-33-0
1,1,1,3,3,3-Hexafluoro-2-propanol	C3H2F6O	920-66-1
Ethoxybenzene	C8H10O	103-73-1
gamma-Butyrolactone	C4H6O2	96-48-0
3-Methylheptane	C8H18	589-81-1
Triethylene glycol dimethyl ether	C8H18O4	112-49-2
Methyl oleate	C19H36O2	112-62-9
Xylene (Isomer not specified)	C8H10	1330-20-7
Perfluoromethylcyclohexane	C7F14	355-02-2
Formamide	CH3NO	75-12-7
n-Undecane	C11H24	1120-21-4
Pentadecane	C15H32	629-62-9
Propylene carbonate	C4H6O3	108-32-7
2,4-Dimethylhexane	C8H18	589-43-5
Tetrahydropyran	C5H10O	142-68-7
Diethylene glycol dimethyl ether	C6H14O3	111-96-6
cis-1,2-Dimethylcyclohexane	C8H16	2207-01-4
Deuterium oxide (Heavy water)	D2O	7789-20-0
N-Methylpyrrolidine	C5H11N	120-94-5
Diiodomethane	CH2I2	75-11-6
Trideuteromethanol	CHD3O	1849-29-2
Perdeuteromethanol	CD4O	811-98-3
Squalane	C30H62	111-01-3
Chlorine	Cl2	7782-50-5
2,2,2-Trifluoroethanol	C2H3F3O	75-89-8
2,2,4,4,6,8,8-Heptamethylnonane	C16H34	4390-04-9
Tributyl phosphate	C12H27O4P	126-73-8
Oxepane	C6H12O	592-90-5

Table B.2 continued.

Component name	**Chemical formula**	**CAS number**
Diethyl succinate	C8H14O4	123-25-1
N-Formylmorpholine	C5H9NO2	4394-85-8
N-Ethylaniline	C8H11N	103-69-5
alpha-Aminotoluene	C7H9N	100-46-9
Hexamethylphosphoric acid triamide	C6H18N3OP	680-31-9
Tripropyl phosphate	C9H21O4P	513-08-6
Cycloheptanone	C7H12O	502-42-1
Iodobenzene	C6H5I	591-50-4
2,3-Dimethylhexane	C8H18	584-94-1
Methylhydrazine	CH6N2	60-34-4
Dinitrogen tetroxide	N2O4	10544-72-6
Dibenzyl ether	C14H14O	103-50-4
1-Bromoheptane	C7H15Br	629-04-9
Cyanoacetic acid methyl ester	C4H5NO2	105-34-0
1,1,7-Trihydroperfluoro-1-heptanol	C7H4F12O	335-99-9
trans-1,2-Dimethylcyclohexane	C8H16	6876-23-9
1,2,3-Propanetriol-triacetate	C9H14O6	102-76-1
1,5-Dimethyl-2-pyrrolidone	C6H11NO	5075-92-3
Carbonic acid diethyl ester	C5H10O3	105-58-8
1,1,3,3-Tetramethyl urea	C5H12N2O	632-22-4
Tricresyl phosphate (Isomer not specified)	C21H21O4P	1330-78-5
1-Chlorohexane	C6H13Cl	544-10-5
N-Methyl-2-piperidone	C6H11NO	931-20-4
N-Methylcaprolactam	C7H13NO	2556-73-2
Tetraethylene glycol dimethyl ether	C10H22O5	143-24-8
Perfluoro (propyl vinyl) ether	C5F10O	1623-05-8
2,6-Dimethylcyclohexanone	C8H14O	2816-57-1
Perfluoro-di-n-butylether	C8F18O	308-48-5
Perfluorodecalin (Isomer not specified)	C10F18	306-94-5

Table B.2 continued.

Component name	**Chemical formula**	**CAS number**
Hexamethyl phosphorous triamide	C6H18N3P	1608-26-0
(Z)-9-Octadecenoic acid ethyl ester	C20H38O2	111-62-6
(Z)-9-Octadecenoic acid butyl ester	C22H42O2	142-77-8
Perfluorotoluene	C7F8	434-64-0
Perfluoro-di-n-pentylether	C10F22O	464-36-8
1,2-Dimethylhydrazine	C2H8N2	540-73-8
Triethylene glycol butylethylether	C12H26O4	184240-60-6
Triethylene glycol monobutyl ether	C10H22O4	143-22-6
Perfluoro-N,N-bis-propyl-1-propanamine	C9F21N	338-83-0
Octanoic acid, 1,2,3-propanetriyl ester	C27H50O6	538-23-8
N-Ethyl-2-pyrrolidone	C6H11NO	2687-91-4
1-Butyl-3-methylimidazolium hexafluorophosphate	C8H15F6N2P	174501-64-5
1-Ethyl-3-methylimidazolium bis(trifluoromethylsulfonyl)imide	C8H11F6N3O4S2	174899-82-2
1-Butyl-3-methylimidazolium bis(trifluoromethylsulfonyl)imide	C10H15F6N3O4S2	174899-83-3
1-Octyl-3-methylimidazolium tetrafluoroborate	C12H23BF4N2	244193-52-0
1-Butyl-3-methylimidazolium nitrate	C8H15N3O3	179075-88-8
1-Ethyl-3-methylimidazolium tetrafluoroborate	C6H11BF4N2	143314-16-3
1-Butyl-3-methylimidazolium tetrafluoroborate	C8H15BF4N2	174501-65-6
Triethylene glycol isopropyl methyl ether	C10H22O4	n.a.
1-Hexyl-3-methylimidazolium bis(trifluoromethylsulfonyl)imide	C12H19F6N3O4S2	382150-50-7
1-Hexyl-3-methylimidazolium tetrafluoroborate	C10H19BF4N2	244193-50-8
Ethylammonium nitrate	C2H8N2O3	22113-86-6
Trihexyl tetradecyl phosphonium chloride	C32H68ClP	258864-54-9

Table B.2 continued.

Component name	**Chemical formula**	**CAS number**
1-Methyl-4-piperidinone	C6H11NO	1445-73-4
1-Butyl-3-methylimidazolium bis(perfluoroethylsulfonyl)imide	C12H15F10N3O4S2	254731-29-8
Trimethyl-butylammonium bis(trifluoromethylsulfonyl)imide	C9H18F6N2O4S2	258273-75-5
Trihexyl tetradecyl phosphonium acetate	C34H71O2P	460092-04-0
Trihexyl tetradecyl phosphonium bis(trifluoromethylsulfonyl)imide	C34H68F6NO4PS2	460092-03-9
1-Hexyl-3-methylpyridinium bis(trifluoromethylsulfonyl)imide	C14H20F6N2O4S2	n.a.
1-(3,3,4,4,5,5,6,6,6-Nonafluorohexyl)-3-methylimidazolium bis(trifluoromethylsulfonyl)imide	C12H10F15N3O4S2	n.a.
2-Hydroxyethyl ammonium acetate	C4H11NO3	54300-24-2
2-Hydroxyethyl ammonium lactate	C5H13NO4	n.a.
Tributylethylphosphonium diethylphosphate	C18H42O4P2	20445-94-7
Tributylmethylphosphonium methylsulfate	C14H33O4PS	69056-62-8
1-Hexyl-3-methylimidazolium tris(pentafluoroethyl)trifluorophosphate	C16H19F18N2P	713512-19-7
1-Propyl-3-methylimidazolium tris(heptafluoropropyl)trifluorophosphate	C16H13F24N2P	n.a.
N-Methyl-2-hydroxyethylammonium propanoate	C6H15NO3	n.a.
N-Methyl-2-hydroxyethylammonium pentanoate	C8H19NO3	n.a.
Selexol	C16H34O8	n.a.
1-Butyl-3-hydrogenimidazolium acetate	C9H16N2O2	n.a.
1-Butyl-3-methylimidazolium tris(pentafluoroethyl)trifluorophosphate	C14H15F18N2P	917762-91-5

Table B.2 continued.

Component name	**Chemical formula**	**CAS number**
1-(Z-Octadec-9-enyl)-3-methylimidazolium bis(trifluoromethylsulfonyl)imide	C24H41F6N3O4S2	1264476-00-7
Octyl-trimethylphosphonium bis(2,4,4-trimethylpentyl)phosphinate	C27H60O2P2	n.a.
Tetrabutylphosphonium caproate	C22H47O2P	n.a.
Tetrabutylphosphonium caprylate	C24H51O2P	n.a.
1-Butyl-3-methylimidazolium caproate	C14H26N2O2	n.a.
1-Butyl-3-methylimidazolium caprylate	C16H30N2O2	n.a.
N,N,N',N'-Tetramethyl-1,3-propanediamine bis(trifluoromethylsulfonyl)imide	C9H19F6N3O4S2	n.a.
Bis(2-dimethylaminoethyl)ether bis(trifluoromethylsulfonyl)imide	C10H21F6N3O5S2	n.a.
Tributylethylphosphonium stearate	C32H67O2P	n.a.

C Supporting Information for Chapter 4.2

C.1 Database

In this chapter, all experimental data on Henry's law constants $H_{ij}(T)$ were taken from the 2021 version of the Dortmund Data Bank (DDB) [86]. The reported $H_{ij}(T)$ values are often derived from various coefficients characterizing gas solubility. Inaccuracies occurring in the conversion of Ostwald coefficients were noticed, and consequently, data from the 2020 version of the DDB were used for these specific instances.

Fig. C.1 shows the temperature distribution of all experimental data on H_{ij}.

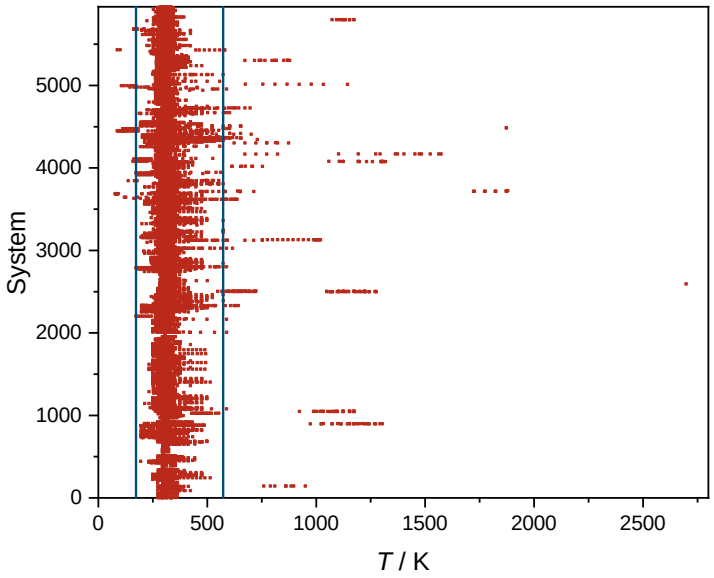


Figure C.1: Temperature distribution of experimental H_{ij} data as reported in the DDB [86]. The order of the binary systems is arbitrary. The focus is placed on data in the temperature range of 173.15 K to 573.15 K, covering 98.5% of the data set. These boundaries are indicated by blue lines.

It shows that the majority of the data were measured at ambient temperature. In this chapter, the focus is placed on the temperature range from 173.15 K to 573.15 K, which includes 98.5% of the total data points.

C.2 PSRK Outliers

In this chapter, PSRK serves both as a benchmark and a source of additional training data for the hybrid MCM. Based on the public parameterization [76], PSRK can predict slightly more than half of the experimental data points (53.74%) from the data set specified above. The results are shown in a parity plot in Fig. C.2.

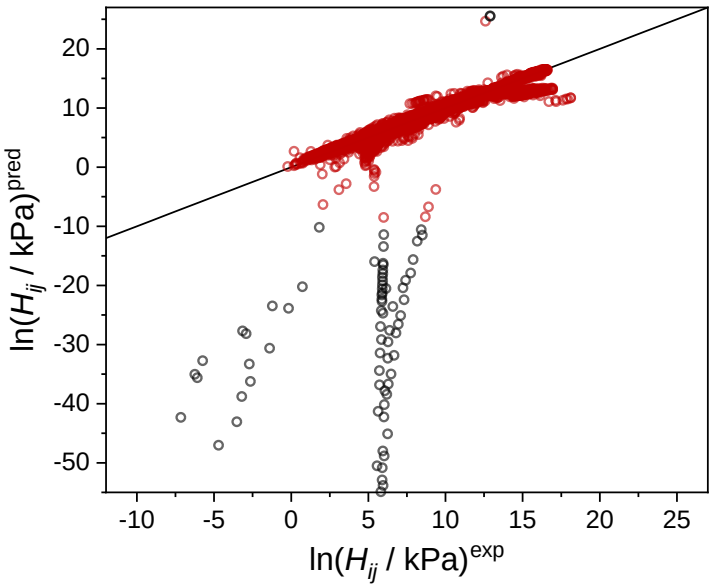


Figure C.2: Parity plot comparing PSRK predictions for $\ln H_{ij}$ in the temperature range 173.15 K - 573.15 K against experimental data (exp) from the DDB [86]. Of the 20,565 experimental data points considered in this chapter, 11,051 can be predicted by PSRK. Among these, 190 were identified as outliers (gray squares, criterion, see Chapter 4.2) and were not used in this chapter; 118 outliers with $\ln H_{ij} < -55$ are not shown for scaling purposes.

Outliers in the PSRK predictions are shown in Fig. C.2. The majority of the detected outliers (183 out of 190) are associated with the solubility of hydrochloric acid (HCl) in water or alcohols, and are probably due to inaccurate group-interaction parameters within the PSRK model.

C.3 Schematic Illustration of the Data-Driven Approach

The workflow of the data-driven matrix completion methods (MCMs) is depicted below, analogously to the hybrid approach illustrated in Fig. 15.

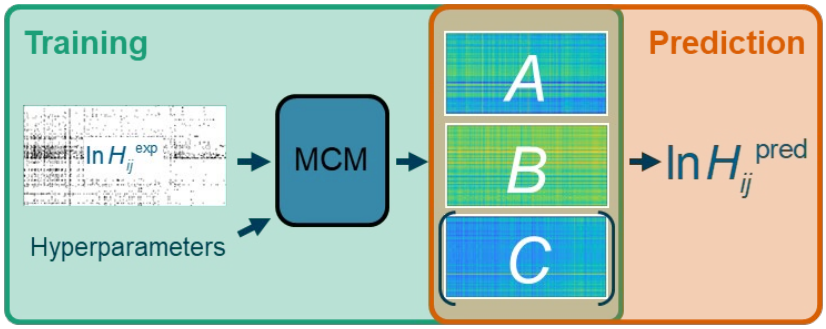


Figure C.3: Schematic illustration of the prediction of $\ln H_{ij}$ with MCM-data. The MCM is trained on experimental data for $\ln H_{ij}$. The inferred features are subsequently used in Eq. (26) to obtain predictions (pred) for the parameters A_{ij}, B_{ij}, and C_{ij} of the function that is used for describing the temperature dependence of $\ln H_{ij}$ (Eqs. (25) or (C.1)), from which the Henry's law constant can be calculated for any temperature.

C.4 MCMs Based on a Two-Parameter Equation

The temperature dependence of the logarithmic Henry's law constant, $\ln H_{ij}(T)$, is commonly modeled using the van't Hoff equation [147–150]:

$$\ln(H_{ij} \,/\, \mathrm{kPa}) = A_{ij} + \frac{B_{ij}}{T \,/\, \mathrm{K}} \tag{C.1}$$

where A_{ij} and B_{ij} are system-specific temperature-independent parameters. Eq. (C.1) is used analogously to Eq. (25) for developing a data-driven and a hybrid MCM, with the parameter matrices A_{ij} and B_{ij} being decomposed as described in Eq. (26). To facilitate comparison and discussion, the abbreviations "2P" for the two-parameter methods and "3P" for the three-parameter methods, respectively, are used in the following.

Fig. C.4 extends Fig. 16 by including error scores (MAE and MSE) for the two-parameter (2P) versions of MCM-data and MCM-hybrid, as defined by Eq. (C.1). To ensure a fair comparison with PSRK, only binary systems predictable by PSRK are considered in Fig. C.4b. However, a direct comparison of PSRK with the MCMs is not trivial. The MCMs are specifically designed to cover binary systems made up of the considered 122 solutes and 399 solvents, allowing them to predict $\ln H_{ij}(T)$ for all potential

solute-solvent combinations. Thus, they are able to describe all 3,297 systems for which experimental data are available in the DDB, cf. Fig. 14. In contrast, PSRK's current parameterization [76] allows it to model only 1,575 of these systems. However, as a group-contribution method, PSRK has the flexibility to model systems beyond those specifically considered in this chapter.

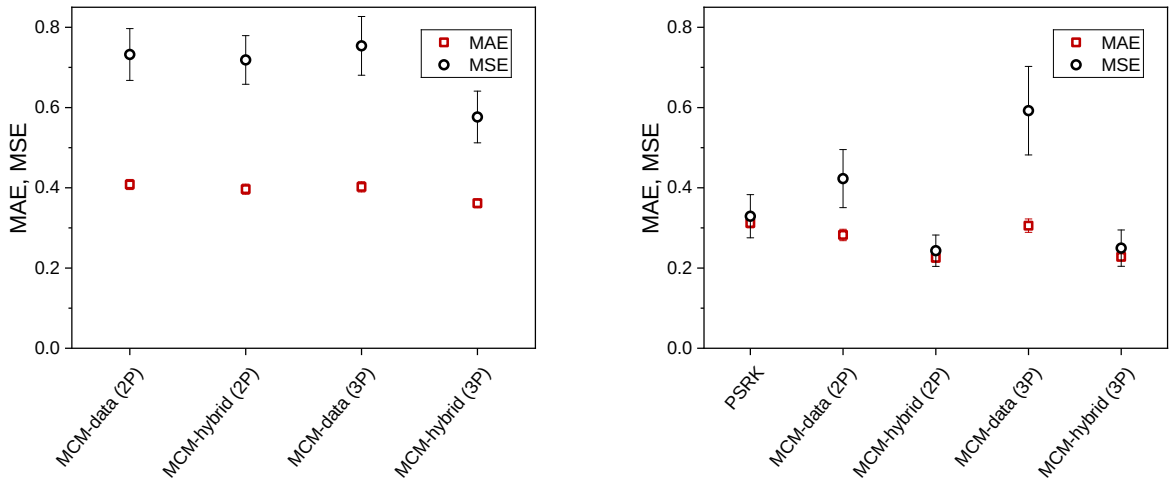
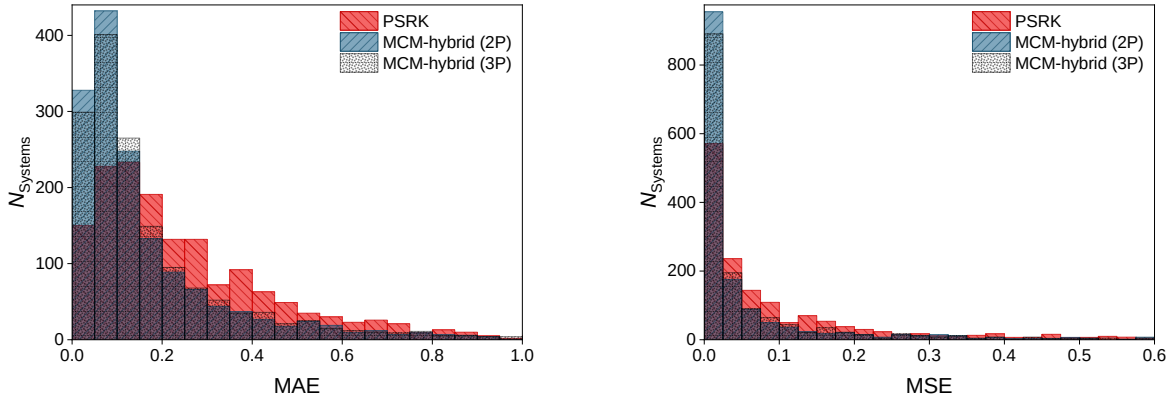


Figure C.4: Mean absolute error (MAE) and mean squared error (MSE) for the prediction of $\ln H_{ij}$ averaged over all binary systems. (a) Comparison of the developed MCMs considering all data from the DDB (3,297 systems). (b) Comparison of PSRK with the developed MCMs considering only systems that can be described by PSRK (1,574 systems).

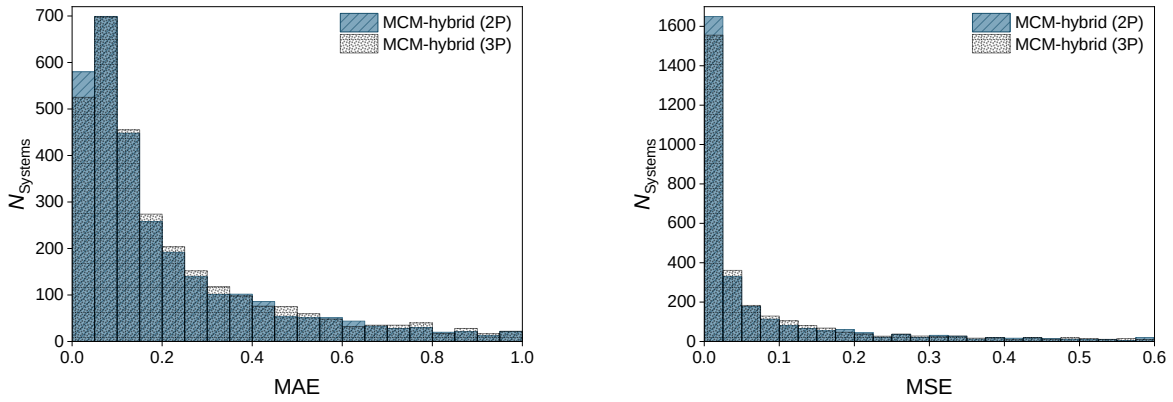
Fig. C.4 illustrates that both 2P-MCMs demonstrate higher predictive accuracies on the data set predictable by PSRK. Furthermore, when relying on a two-parameter equation, the hybrid approach outperforms both the data-driven MCMs and PSRK.

Fig. C.4b reveals no significant difference in the performance of MCM-hybrid (2P) and MCM-hybrid (3P) against PSRK. However, MCM-hybrid (2P) demonstrates limitations in accurately describing systems beyond PSRK's predictive scope, highlighting its reliance on additional information provided during the pretraining step, cf. Fig. C.4a. The performance of the hybrid MCMs is further analyzed in Figs. C.5 and C.6, where the error scores MAE and MSE for each binary system are illustrated as histograms.



(a) **(b)**

Figure C.5: Histogram representation of the MAE and MSE for the predictions of $\ln H_{ij}$ with PSRK, MCM-hybrid (2P), and MCM-hybrid (3P) considering only systems that can be described by PSRK (1,574 systems). N_{Systems} is the number of binary systems. (a) MAE. The shown interval in the histogram contains 96.32% (PSRK), 96.63% (MCM-hybrid (2P)), and 96.63% (MCM-hybrid (3P)) of all considered binary systems. (b) MSE. The shown interval in the histogram contains 93.58% (PSRK), 94.41% (MCM-hybrid (2P)), and 94.03% (MCM-hybrid (3P)) of all considered binary systems.



(a) **(b)**

Figure C.6: Histogram representation of the MAE and MSE for the predictions of $\ln H_{ij}$ with MCM-hybrid (2P), and MCM-hybrid (3P) considering all data from the DDB (3,297 systems). N_{Systems} is the number of binary systems. (a) MAE. The shown interval in the histogram contains 94.51% (MCM-hybrid (2P)), and 95.54% (MCM-hybrid (3P)) of all considered binary systems. (b) MSE. The shown interval in the histogram contains 92.45% (MCM-hybrid (2P)), and 93.27% (MCM-hybrid (3P)) of all considered binary systems.

Both MAE and MSE reveal the same trend in predictive accuracy: MCM-hybrid (2P) achieves highly accurate predictions across a variety of systems, slightly outperforming MCM-hybrid (3P). This suggests the difference in performance between MCM-hybrid (2P) and MCM-hybrid (3P), as observed in Fig. C.4a, should not be interpreted as a lack of overall efficacy of the 2P approach. Rather, it is primarily the impact of significant outliers that cause the higher error scores.

C.5 Enthalpy of Absorption

The enthalpy of absorption of the solute i in the pure solvent j can be obtained from the temperature derivative of the Henry's law constant [72, 151, 152]:

$$\Delta h_{ij}^{\mathrm{abs}} = -RT^2 \cdot \frac{\partial \ln H_{ij}}{\partial T} \tag{C.2}$$

or,

$$\Delta h_{ij}^{\mathrm{abs}} = R \cdot \frac{\partial \ln H_{ij}}{\partial \frac{1}{T}} \tag{C.3}$$

with

$$\Delta h_{ij}^{\mathrm{abs}} = h_{ij}^{\infty} - h_i^{\mathrm{id}} \tag{C.4}$$

where R is the universal gas constant, h_{ij}^{∞} is the partial molar enthalpy of solute i infinitely diluted in solvent j, and h_i^{id} is the molar enthalpy of component i in the ideal gas state. Substituting $\ln H_{ij}(T)$ with the two-parameter equation (Eq. (C.1)) or the three-parameter equation (Eq. (25)), yields the following expressions for $\Delta h_{ij}^{\mathrm{abs}}$:

$$\Delta h_{ij}^{\mathrm{abs,\,2P}} = R \cdot B_{ij} \tag{C.5}$$

$$\Delta h_{ij}^{\mathrm{abs,\,3P}} = R \cdot (B_{ij} - C_{ij}(T - T_0)) \tag{C.6}$$

where B_{ij} and C_{ij} are the parameters of the respective equations, and $T_0 = 298$ K is the chosen reference temperature. Due to the scarcity of experimental data on $\Delta h_{ij}^{\mathrm{abs}}$, these values could not be utilized for training the developed MCMs or for systematically evaluating their predictive accuracy. However, Fig. C.7 shows predicted $\Delta h_{ij}^{\mathrm{abs}}$ for the six examples discussed in Fig. 18.

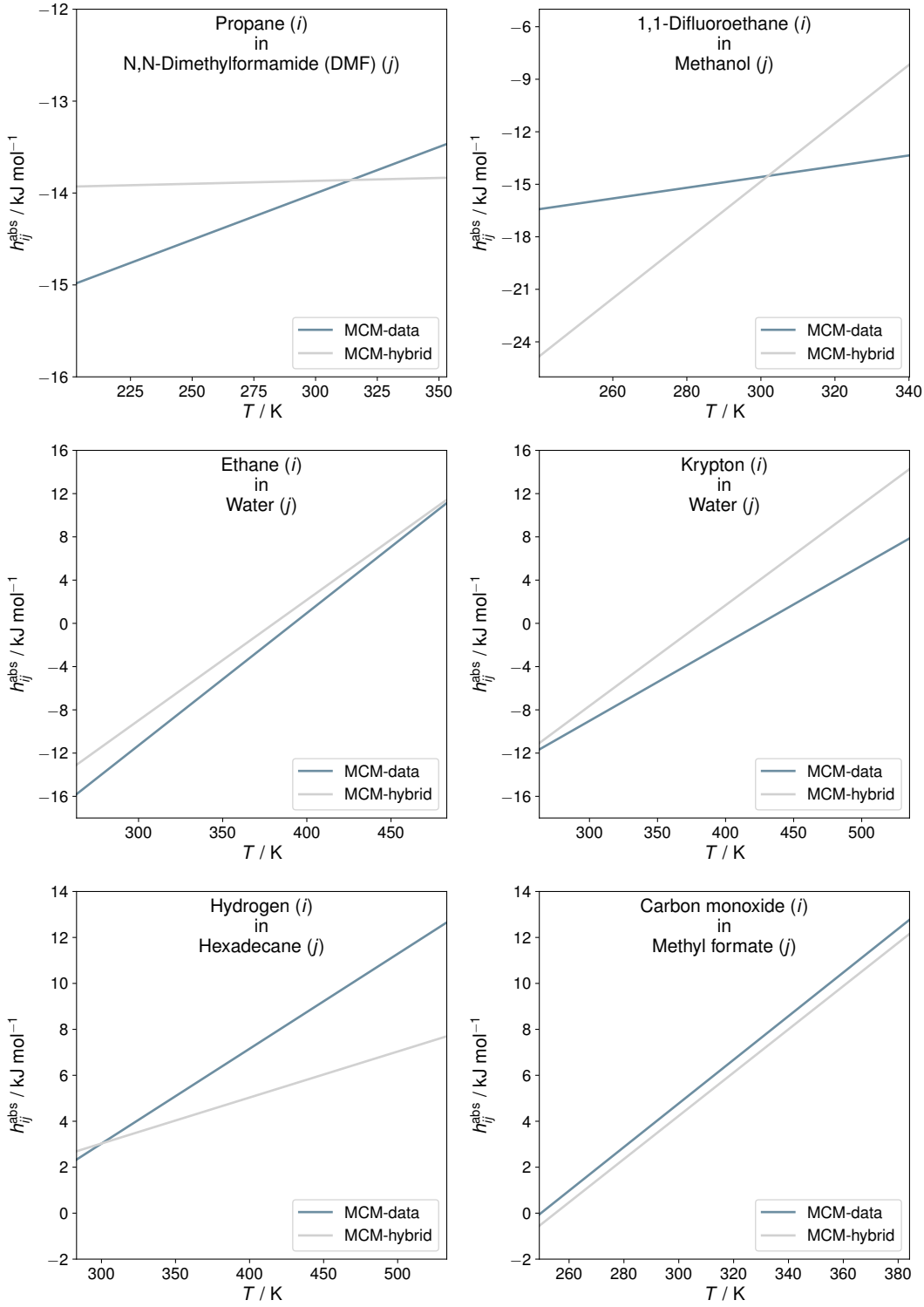


Figure C.7: Predicted enthalpies of absorption $\Delta h_{ij}^{\mathrm{abs}}$ in six binary systems using MCM-data (blue), and MCM-hybrid (grey). The results of the MCMs are true predictions obtained using leave-one-out analysis.

D Supporting Information for Chapter 4.4

D.1 Semiempirical Models

In this section, the considered semiempirical models studied in the present thesis are briefly presented. Further, it is shown exactly how D_{ij}^{∞} is calculated in each of these cases from some pure-component properties of the solutes and solvents. The pure-component properties needed for this purpose were calculated for the studied temperature T = 298.15 K by DIPPR correlations, which are provided in the DIPPR database [113]. For the solutes i and solvents j, these include, depending on the model, some combination of the molar masses M_i and M_j, the parachors[3] P_i and P_j, and the saturated liquid phase molar volumes $\tilde{v}_i$ and $\tilde{v}_j$ at the respective normal boiling temperatures of the solute and solvent, as well as the viscosity η_j of the solute.

With the exception of SEGWE, the semiempirical models considered here need information on the saturated liquid phase molar volume $\tilde{v}_i$ of the solute i at its normal boiling temperature. However, for carbon dioxide this value is not defined since its triple point pressure is above the ambient pressure; therefore, the liquid molar volume at the triple point was used instead here. Similarly, also for perylene and 3-hydroxyaniline $\tilde{v}_i$ at the normal boiling temperature cannot be measured since both components decompose before reaching the respective temperatures; therefore, a hypothetical value for $\tilde{v}_i$ has been used for these components, which was calculated with the group-contribution method of Schröder [154].

While the four semiempirical models have been developed as general-purpose correlations that aim at describing a diverse set of mixtures and components, there are still some restrictions in the scope of these models, which are briefly mentioned here. All authors have limited their models to moderate viscosities and have excluded data for viscous solvents (e.g., polymers) from their training sets. Further, none of the semiempirical models were trained on data of mixtures containing electrolytes, i.e., neither

[3] The parachor is used here as defined by Quayle: $P_i = \sqrt[4]{\gamma_i} v_i$, where γ_i and v_i are the surface tension and liquid molar volume of pure component i at the studied temperature, respectively [153].

mixtures with salts as solutes nor with ionic liquids as solutes or solvents should be expected to be predicted with high accuracy.

D.1.1 Wilke and Chang, 1955

One of the first widely applicable correlations for diffusion coefficients in liquids was developed by Wilke and Chang [94]. According to the model of Wilke and Chang, D_{ij}^{∞} is calculated by:

$$\left(\frac{D_{ij}^{\infty}}{\mathrm{m^2/s}}\right) = 7.4 \times 10^{-12} \sqrt{\phi_j \left(\frac{M_j}{\mathrm{g/mol}}\right)} \frac{1}{\left(\frac{\tilde{v}_i}{\mathrm{cm^3/mol}}\right)^{0.6}} \frac{\left(\frac{T}{\mathrm{K}}\right)}{\left(\frac{\eta_j}{\mathrm{mPa\ s}}\right)} \tag{D.1}$$

where ϕ_j is a solvent-specific factor, which was introduced to improve the description of diffusion coefficients in associating solvents; for some common solvents, values for ϕ_j have been reported [94]. However, in this thesis, values for ϕ_j were fitted for each solvent individually to experimental D_{ij}^{∞} from the database (cf. Section 4.4.3.2.4).

D.1.2 Reddy and Doraiswamy, 1967

Reddy and Doraiswamy sought to improve on the Wilke-Chang correlation by eliminating the factor ϕ_j and considering the molar volume $\tilde{v}_j$ of the solvent instead [95]. They also changed the exponent of both $\tilde{v}_i$ and $\tilde{v}_j$ to $\frac{1}{3}$, an idea that was previously introduced by Scheibel [155], resulting in Equation D.2:

$$\left(\frac{D_{ij}^{\infty}}{\mathrm{m^2/s}}\right) = K_{\mathrm{RS}} \frac{\sqrt{\frac{M_j}{\mathrm{g/mol}}}}{\sqrt[3]{\left(\frac{\tilde{v}_i}{\mathrm{cm^3/mol}}\right)\left(\frac{\tilde{v}_j}{\mathrm{cm^3/mol}}\right)}} \frac{\left(\frac{T}{\mathrm{K}}\right)}{\left(\frac{\eta_j}{\mathrm{mPa\ s}}\right)} \tag{D.2}$$

The empirical constant K_{RS} depends on the ratio of $\tilde{v}_i$ to $\tilde{v}_j$:

$$K_{\mathrm{RS}} = \begin{cases} 10 \times 10^{-12}, & \text{for } \frac{\tilde{v}_j}{\tilde{v}_i} \leq 1.5 \\ 8.5 \times 10^{-12}, & \text{for } \frac{\tilde{v}_j}{\tilde{v}_i} > 1.5 \end{cases} \tag{D.3}$$

D.1.3 Tyn and Calus, 1975

Tyn and Calus found that the ratio of the parachors P_i and P_j correlates strongly with D_{ij}^{∞} [96], and therefore proposed the following equation:

$$\left(\frac{D_{ij}^{\infty}}{\mathrm{m^2/s}}\right) = 8.93 \times 10^{-12} \sqrt[6]{\frac{\left(\frac{\tilde{v}_i}{\mathrm{cm^3/mol}}\right)}{\left(\frac{\tilde{v}_j}{\mathrm{cm^3/mol}}\right)^2}} \left(\frac{P_j}{P_i}\right)^{0.6} \frac{\left(\frac{T}{\mathrm{K}}\right)}{\left(\frac{\eta_j}{\mathrm{mPa\ s}}\right)} \tag{D.4}$$

The Tyn and Calus model is subject to the following restrictions [96]:

- For the solute water, the authors suggest that water should be treated as a dimer, i.e., the values of $\tilde{v}_i$ and P_i should be doubled. In this thesis, the values $\tilde{v}_{water}$ = 37.4 cm^3/mol and P_{water} = 105.2 cm^3 g$^{1/4}$/(s$^{1/2}$ mol) for the water dimer have been used, as recommended by Poling [99].
- When the solute is an organic acid, the dimer value of $2\tilde{v}_i$ and $2P_i$ should be used in solvents other than water, methanol, and butanol. In the present thesis, this suggestion has been followed.
- For nonpolar solutes in monohydroxy alcohol solvents, the values of $\tilde{v}_j$ and P_j should be multiplied by the factor $8\eta_j$, with the solvent viscosity η_j in units of mPa s, which was done accordingly in the present thesis.

D.1.4 SEGWE (Stokes-Einstein Gierer-Wirtz Estimation)

In a recent work of Evans et al., the Stokes-Einstein equation [98] was extended by introducing the Gierer-Wirtz [156] correction to loosen the assumption of the Stokes-Einstein theory that the solvent is a continuum fluid [97]. Consequently, they named their model SEGWE (Stokes-Einstein Gierer-Wirtz Estimation), which calculates D_{ij}^{∞} as:

$$D_{ij}^{\infty} = \frac{k_\mathrm{B}}{6\pi} \frac{\left(\frac{3\alpha}{2} + \frac{1}{1+\alpha}\right)}{\sqrt[3]{\frac{3M_i}{4\pi\varrho_\mathrm{eff}N_\mathrm{A}}}} \frac{T}{\eta_j} \tag{D.5}$$

where ϱ_eff is the effective density and α is the ratio of the solvent and solute radii, r_j and r_i, respectively. Further, k_B and N_A are the Boltzmann and Avogadro constants, respectively. Assuming that all molecules are hard spheres, α can also be expressed in terms of the molar masses M_j and M_i:

$$\alpha = \frac{r_j}{r_i} = \sqrt[3]{\frac{M_j}{M_i}} \tag{D.6}$$

The effective density ϱ_{eff}, which can be considered either as a solvent-specific parameter or fitted to a global value, was fitted by the original authors to diffusion coefficient data at 25 °C for 109 combinations of 44 solutes and 5 solvents, yielding a global value of 619 kg/m^3 [97].

In the present thesis, ϱ_{eff} is used as a solvent-specific parameter, which has been fitted individually to the respective data on D_{ij}^{∞} for each solvent from the database; as described above, a leave-one-out strategy was thereby followed (cf. Section 4.4.3.2.4).

D.1.5 Effect of Fitting the Model Parameters With a Leave-One-Out Strategy

Both the Wilke-Chang and SEGWE models contain a solvent-specific fit parameter, called ϕ_j and $\varrho_{\text{eff},j}$, respectively. For a fair comparison to the MCMs, these were fitted to the database using a leave-one-out strategy: i.e., for the prediction of each experimental D_{ij}^{∞}, a $\phi_j^{(i)}$ (or $\varrho_{\text{eff},j}^{(i)}$) was fitted to all available experimental data in that particular solvent *minus* the data point $i + j$ that is to be predicted. The optimum $\phi_j^{(i),*}$ was chosen for the minimum in the rRMSE:

$$\phi_j^{(i),*} = \underset{\phi_j^{(i)}}{\arg\min} \sum_{k \neq i} \left(\frac{D_{kj}^{\infty,\text{pred}}\left(\phi_j^{(i)}\right) - D_{kj}^{\infty,\text{exp}}}{D_{kj}^{\infty,\text{exp}}} \right)^2 \tag{D.7}$$

However, it is also possible to apply the Wilke-Chang and SEGWE models in a purely predictive manner: for Wilke-Chang this means using the (few) parameter values of ϕ_j supplied by the original authors, for SEGWE the global value $\varrho_{\text{eff}} = 619$ kg/m^3 is used.

For both models, there is only a small difference in the overall performance when comparing the purely predictive approach to that with the fitted parameter. The effect is shown in Fig. D.1. For SEGWE, the rMAE and rRMSE decrease from 0.213 and 0.285 in the predictive approach to 0.193 and 0.276 in the fitted approach, respectively. For Wilke-Chang, the rMAE decreases from 0.227 to 0.209, while, surprisingly, the rRMSE slightly increases from 0.304 to 0.314. This paradoxical effect is due to the large number of solvents in which data is available only for very little mixtures (i.e. solvents that have been measured in combinations with few solutes). In such cases, the leave-one-out strategy will lead to a good fit of ϕ_j to the (limited) available data, while the left-out point may therefore be grossly mispredicted, resulting in a high rRMSE.

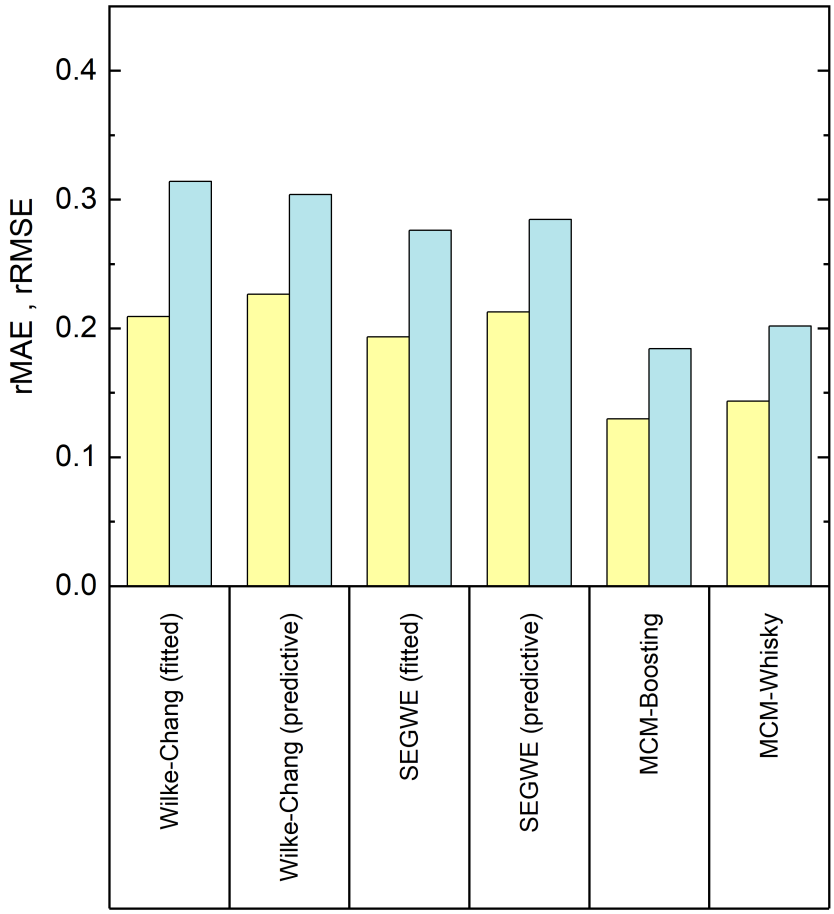


Figure D.1: Relative mean absolute error (rMAE, yellow) and relative root mean squared error (rRMSE, blue) of the predicted D_{ij}^{∞} for the experimental data from the reduced database. The developed MCMs are compared to the semiempirical models Wilke-Chang and SEGWE in two variants: a purely predictive one and one that was fitted to the database using a leave-one-out strategy.

D.1.6 Mixtures Poorly Described by Semiempirical Models

In this section, a closer look is taken at those mixtures from the database for which D_{ij}^{∞} is only poorly described by the semiempirical models, and an attempt is made to specify the groups of solutes and solvents for which this is the case. The focus is thereby placed on SEGWE, while the other models are also briefly touched upon.

For discussing the performance of SEGWE in detail, Fig. 28, which shows the residuals of the SEGWE predictions from the experimental data, is referred to. One solute that SEGWE is apparently struggling to describe accurately is water (solute i = 27, cf. Fig. 28). In the reduced database, there are eight mixtures with the solute water; the relative deviations of the SEGWE predictions from the experimental data for D_{ij}^{∞} for these eight mixtures are shown in Fig. D.2.

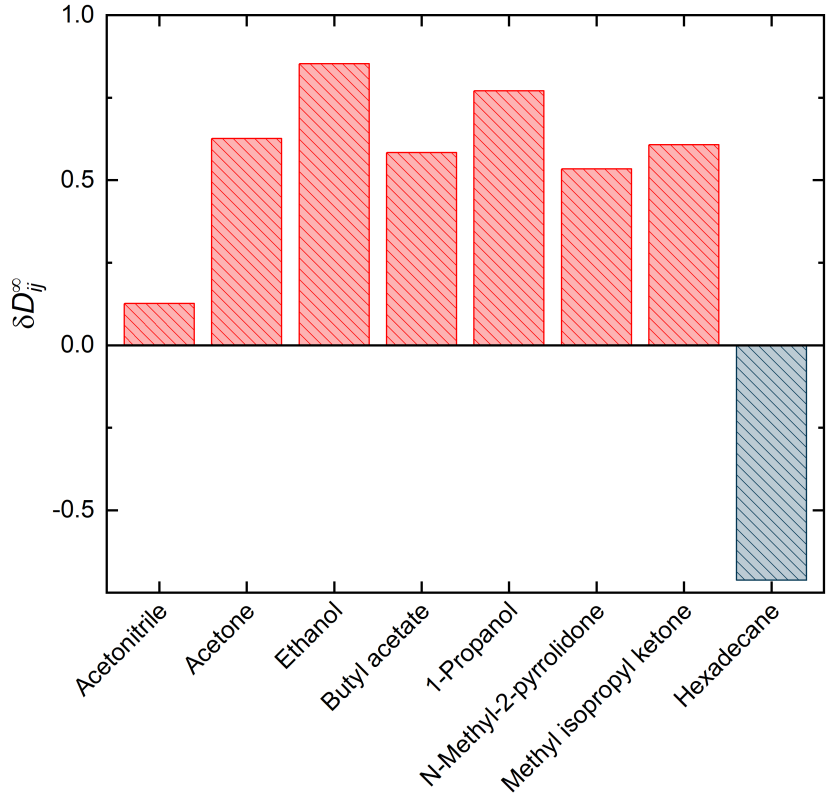


Figure D.2: Relative deviations $\delta D_{ij}^{\infty} = \left(D_{ij}^{\infty,\mathrm{pred}} - D_{ij}^{\infty,\mathrm{exp}}\right)/D_{ij}^{\infty,\mathrm{exp}}$ of the SEGWE predictions for D_{ij}^{∞} of the solute water in different solvents from the experimental data from the reduced database.

The largest positive relative deviations are found for mixtures in which strong hydrogen bonding occurs, namely the mixtures (water + ethanol) and (water + 1-propanol). Slightly smaller, but still large positive relative deviations are found for mixtures of water with solvents in which weaker hydrogen bonds are formed (acetone, butyl acetate, *N*-methyl-2-pyrrolidone and methyl isopropyl ketone, cf. Fig. D.2). This is not astonishing as the developers of SEGWE have explicitly excluded data for mixtures with "aggregating components" in the development of SEGWE [97]. Aggregation leads to lower diffusion coefficients; an effect which is not described by SEGWE, which, as a consequence, overpredicts D_{ij}^{∞} in such mixtures, cf. Fig. D.2. High positive relative deviations of the SEGWE predictions from the experimental data are also found for many other hydrogen bonding systems in the database.

Furthermore, SEGWE mispredicts D_{ij}^{∞} in mixtures where the molecular mass in relation to the molecule size strongly differs between both components. This is in particular the case if one of the components contains heavy atoms, and the other does not. The reason for this is that in the development of SEGWE, it was assumed that both solute and solvent can be modeled as hard spheres, and that both spheres have an equal ratio of mass to volume – the so-called effective density ϱ_{eff} of the mixture.

An instructive example for this case is the result for the solute carbon dioxide (i = 39) in Fig. 28. Carbon dioxide has a relatively large molecular mass in relation to its small molecular volume, which leads to a rather high effective density compared to, e.g., typical organic solvents. Accordingly, SEGWE significantly underestimates D_{ij}^{∞} for basically all mixtures with carbon dioxide from the reduced database (cf. Fig. 28), and even for all mixtures with carbon dioxide from the full database (not shown here).

Two other examples for solutes in the database with rather high effective densities are methyl iodide (i = 19), which is due to the heavy iodine atom, and the fully fluorinated hexafluorobenzene (i = 30); SEGWE also underestimates the diffusion in all mixtures containing these two solutes. Returning to Fig. D.2 as a last example, the significant underestimation of the experimental D_{ij}^{∞} in the mixture (water + hexadecane) can likewise be explained by the higher effective density of water in relation to that of hexadecane (and the absence of significant attractive forces in the mixture to counteract this effect).

Finally, the limitations of the models of Wilke and Chang [94], Reddy and Doraiswamy [95], and Tyn and Calus [96] are briefly touched upon. Due to their similar nature they are all subject to similar restrictions, so that they will be discussed together here. Despite the original authors' intention to provide general-purpose correlations that work in nonpolar and polar mixtures alike, all three models have been found to struggle significantly with hydrogen bonding mixtures (as it is also the case for SEGWE). Hence, they overpredict D_{ij}^{∞} for hydrogen bonding solvents, such as methanol, ethanol and 1-propanol. Further, the Wilke-Chang model is inaccurate in the prediction of the diffusion of water in organic solvents, which has been described before in the literature [157]. Accordingly, a significant overestimation of D_{ij}^{∞} by the Wilke-Chang model can be observed for nearly all mixtures from the reduced database in which water is the solute, with the exception of the mixture (water + hexadecane). This trend is not observed for the models of Tyn and Calus or Reddy and Doraiswamy.

Lastly, it should be noted that MCMs can be used to identify such systematic deviations in the predictions of (semiempirical) models, and that MCMs can also predict them, which is used in the hybrid MCM based on "boosting" for improving the performance of the semiempirical models, cf. Fig. 27.

D.2 Maximum Errors in the Predictive Performance of the Studied Models

In Fig. D.3, the relative maximum absolute errors, defined by Equation D.8, of the predictions for D_{ij}^{∞} with the four semiempirical models and the three MCMs studied in

this chapter on the reduced database are shown. Similar results to those in Fig. 27 are found, namely that the performance of the data-driven MCM suffers from some drastic mispredictions (leading to the high relative maximum absolute error seen here), and that both hybrid MCMs outperform the semiempirical models in this statistic as well. Again, it is found that MCM-Boosting performs slightly better than MCM-Whisky in terms of the relative maximum absolute error.

$$\text{relative maximum absolute error} = \max_{i,j} \left| \frac{D_{ij}^{\infty,\text{pred}} - D_{ij}^{\infty,\text{exp}}}{D_{ij}^{\infty,\text{exp}}} \right| \tag{D.8}$$

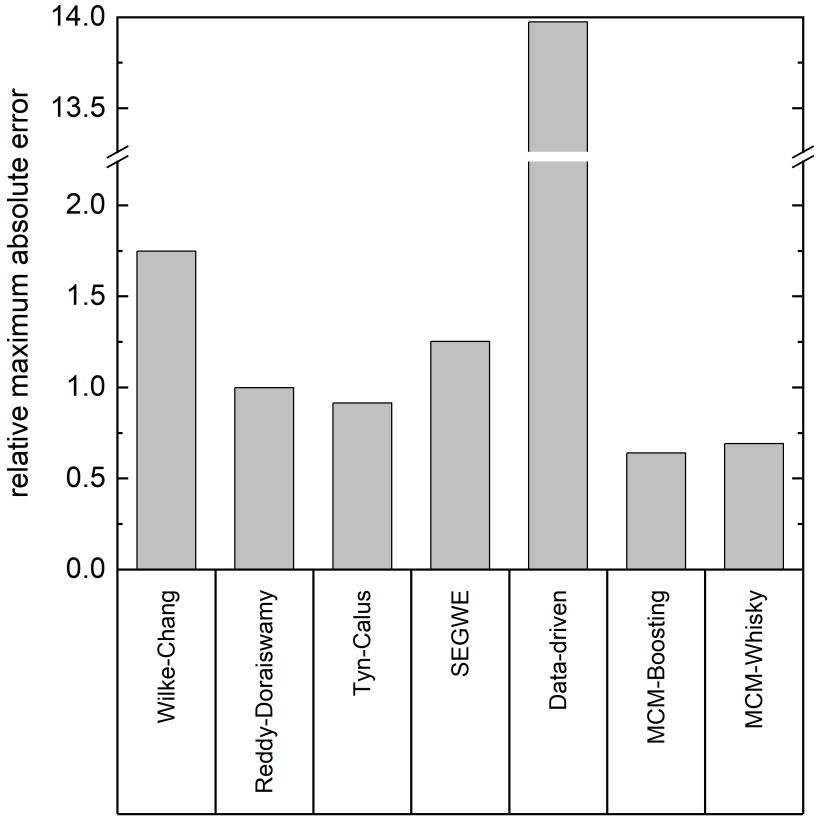


Figure D.3: Relative maximum absolute error of the predicted D_{ij}^{∞} with the studied semiempirical models and the developed MCMs for the experimental data from the reduced database.

D.3 Complete Predictions from MCM-Whisky

Analogous to the MCM-Boosting results in Fig. 31, the completed D_{ij}^{∞} matrix from the MCM-Whisky predictions, together with the uncertainties of those predictions, are shown in Fig. D.4.

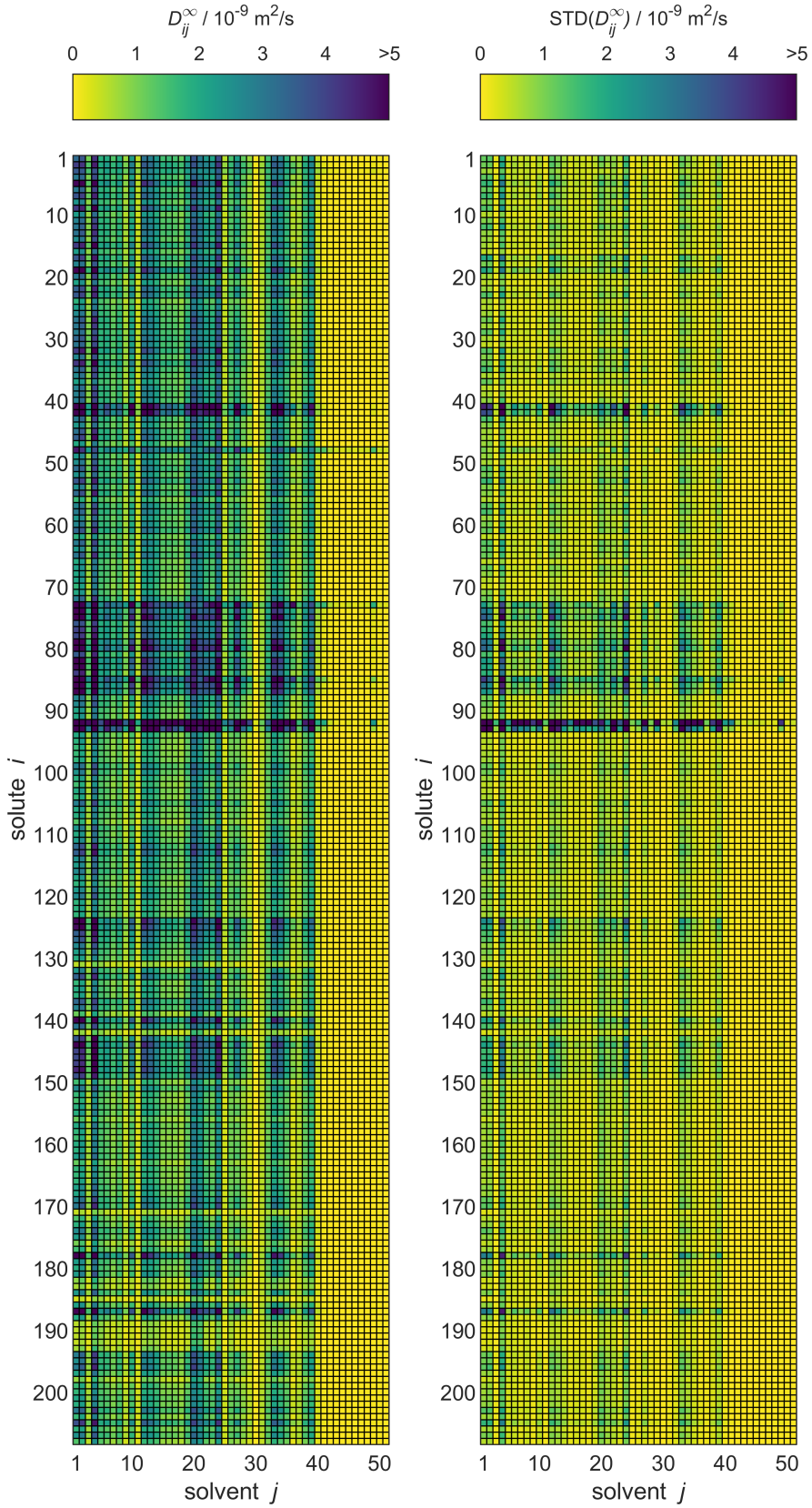


Figure D.4: Predictions of D_{ij}^{∞} by MCM-Whisky (left) and the uncertainties of the predictions (right) for all solutes i and solvents j (identified by numbers, see Table D.1) from the full database. The color code indicates the values of D_{ij}^{∞}.

D.4 Supplementary Tabular Files

Additional supplementary information is provided in a machine readable format in the form of .csv files in Ref. [83]. The data are provided in two separate folders, named "full" and "reduced", representing the full database and the reduced database. In each folder, the following files are found:

- *Boosting_Predictions.csv*: Here, the predictions of D_{ij}^{∞} with the hybrid MCM "MCM-Boosting" (cf. Section 4.4.3.2.2) are reported. The results were obtained here after training the model on all data on D_{ij}^{∞} in the full (reduced) database. The predictions are listed for all 10,608 (1,035) solute-solvent combinations and include a large number of novel data points on D_{ij}^{∞}. In the same table, the model uncertainty of each predicted D_{ij}^{∞} is listed next to the predicted value in the form of standard deviations.

- *Boosting_LV_Solutes.csv* and *Boosting_LV_Solvents.csv*: Here, the results of the training of MCM-Boosting on the full (reduced) database of D_{ij}^{∞} are reported, which are the feature vectors $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$ of the solutes and solvents, respectively. The length of the feature vectors is $K = 2$.

- *Whisky_Predictions.csv*: Here, the predictions of D_{ij}^{∞} with the hybrid MCM "MCM-Whisky" (cf. Section 4.4.3.2.2) are reported. The results were obtained here after training the model on all data on D_{ij}^{∞} in the full (reduced) database. The predictions are listed for all 10,608 (1,035) solute-solvent combinations and include a large number of novel data points on D_{ij}^{∞}. In the same table, the model uncertainty of each predicted D_{ij}^{∞} is listed next to the predicted value in the form of standard deviations.

- *Whisky_LV_Solutes.csv* and *Whisky_LV_Solvents.csv*: Here, the results of the training of MCM-Whisky on the full (reduced) database of D_{ij}^{∞} are reported, which are the feature vectors $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$ of the solutes and solvents, respectively. The length of the feature vectors is $K = 2$.

An excerpt of this information is also provided in written form in Tables D.1-D.3.

D.5 Tabular Material

Table D.1: Table of all components, subdivided into solutes and solvents, encountered in the database on D_{ij}^{∞} developed in Ref. [83]. All components are listed by their consecutive number, as used in all figures throughout this chapter, together with their DDB identification number.

Cons. No.	DDB No.	Name	Cons. No.	DDB No.	Name
Solutes					
1	2	Acetamide	105	3063	L-Ascorbic acid
2	3	Acetonitrile	106	3215	4-Hydroxy-3-methoxybenzaldehyde
3	4	Acetone	107	3258	2,2-Bis(hydroxymethyl)-1,3-propanediol
4	8	1,2-Ethanediol	108	3347	D-(+)-Saccharose
5	11	Ethanol	109	3410	1,3,5-Triisopropylbenzene
6	12	Diethyl ether	110	3468	DL-Phenylalanine
7	15	Formic acid	111	3523	1,4-Diaminobenzene
8	17	Aniline	112	3715	Benzenesulfonic acid
9	21	Ethyl acetate	113	3717	p-Toluenesulfonic acid
10	24	Benzyl alcohol	114	3724	L-Alanine
11	25	Ethylbenzene	115	3725	L-Serine
12	26	Bromobenzene	116	3729	Glycine
13	27	Chlorobenzene	117	3731	L-(+)-Aspartic acid
14	30	Nitrobenzene	118	3732	L-Glutamic acid
15	31	Benzene	119	3865	Piperazine
16	39	1-Butanol	120	3988	beta-Alanine
17	40	2-Butanone	121	3989	4-Aminobutyric acid
18	41	n-Butane	122	3990	5-Aminovaleric acid
19	46	Butyl chloride	123	3991	6-Aminohexanoic acid
20	47	Chloroform	124	4490	Potassium thiocyanate
21	49	3-Methylphenol	125	4577	Potassium chloride
22	50	Cyclohexane	126	4591	Cadmium chloride
23	72	N,N-Dimethylformamide	127	4592	Nickel chloride
24	77	2,6-Dimethylpyridine	128	4596	Ferrocene
25	78	Dodecane	129	4707	(+-)-alpha-Aminobutyric acid
26	79	Benzaldehyde	130	4708	alpha-Aminoisobutanoic acid
27	80	Butyl acetate	131	4771	Buckminsterfullerene
28	84	Acetic acid	132	4776	2-Acetoxy benzoic acid
29	85	Furfural	133	4792	Sodium nitrate
30	89	Hexane	134	4795	D-Mannose
31	91	Heptane	135	4801	!D-Xylose
32	99	Methyl iodide	136	4817	1,2,6-Hexanetriol
33	108	1-Methylnaphthalene	137	4911	Sodium chloride
34	110	Methanol	138	4955	Magnesium chloride
35	112	3-Methylpentane	139	4960	Magnesium sulfate
36	123	Naphthalene	140	4965	Potassium nitrite
37	129	1-Octene	141	5261	1,2-Ethanediol-D2 (deuterioglycol)
38	138	Phenol	142	5949	3,4-Dihydroxy benzoic acid
39	140	1-Propanol	143	6317	Iron(III) sulfate
40	141	Propionic acid	144	6319	Ammonium chloride
41	145	Nitric acid	145	6325	Ammonium sulfate
42	146	Hydrogen chloride	146	6326	Lead nitrate
43	147	Salicylic acid methyl ester	147	6353	Sodium perchlorate
44	153	tert-Butanol	148	6355	Potassium chlorate
45	157	Tetrachloromethane	149	6372	Sodium thiocyanate
46	161	Toluene	150	6465	N-Acetyl-p-aminophenol
47	168	Trichloroethylene	151	6529	Di-tert-butylsulfide
48	174	Water	152	7467	Titanium tetra-tert.butyloxide
49	230	Glycerol	153	7533	15-Crown-5 (15C5)
50	235	Butyric acid	154	7847	L-Valine
51	237	Propane	155	7848	L-Isoleucine
52	250	Cyclohexanone	156	7852	L-Tryptophane
53	269	Caprylic acid	157	7949	L-Cystine
54	284	N-Methyl-2-pyrrolidone	158	9329	7-Aminoheptanoic acid

Table D.1 continued.

Cons. No.	DDB No.	Name	Cons. No.	DDB No.	Name
55	297	Hexafluorobenzene	159	10334	Tris(2,4-pentanedionato)chromium
56	308	2-Methyl-2,4-pentanediol	160	10571	Phenylphosphonic acid
57	322	o-Xylene	161	11004	D-Galactose
58	367	2,3-Dimethylbutane	162	11201	Sodium caprylate
59	372	Acetophenone	163	11202	Sodium dodecyl sulfate
60	425	Benzoic acid	164	11722	L-Threonine
61	430	Methyl isopropyl ketone	165	12706	D-Glucose
62	516	Hexadecane	166	13599	L-Lysine
63	546	Monoethanolamine	167	16447	1-Ethyl-3-methylimidazolium bis(trifluoromethylsulfonyl)imide
64	598	Trifluoroacetic acid	168	16583	1-Butyl-3-methylimidazolium bis(trifluoromethylsulfonyl)imide
65	750	p-Chlorotoluene	169	16584	1-Ethyl-3-methylimidazolium ethylsulfate
66	766	1,2-Dihydroxybenzene	170	16731	Cadmium perchlorate
67	809	2-Methoxyphenol	171	17118	(-)-Epicatechin
68	810	o-Chlorophenol	172	17231	Calcium-L-lactate
69	812	p-Chlorophenol	173	17273	D-(-)-Arabinose
70	817	1,3-Dihydroxybenzene	174	17617	tert-Butan(ol-D)
71	894	2,2”-Diethanolamine (DEA)	175	18690	Monosodium glutamate
72	925	Anthracene	176	18840	Lysozyme
73	1050	Carbon dioxide	177	18842	L-3,4-Dihydroxyphenylalanine
74	1051	Methane	178	18845	1-Butyl-3-methylimidazolium methylsulfate
75	1052	Oxygen	179	18857	Monosodium L-aspartate
76	1053	Ethylene	180	19687	L-Arginine
77	1054	Ethane	181	20036	1-Butyl-3-methylimidazolium octyl sulfate
78	1055	Propylene	182	20046	alpha-Cyclodextrin
79	1056	Nitrogen	183	20047	beta-Cyclodextrin
80	1058	Argon	184	22696	5-Hydroxymethylfurfural
81	1059	Chlorine	185	23228	Isoquercitrin
82	1060	Krypton	186	23325	(.+-.)-.beta.-Aminobutyric acid
83	1061	Dinitrogen monoxide	187	26695	[EMIM] methylsulfate
84	1062	Xenon	188	26828	Platinum (II) acetylacetonate
85	1063	Hydrogen	189	33333	Gallic acid monohydrate
86	1064	Ethyne	190	33334	(+)-Catechin hydrate
87	1065	Hydrogen sulfide	191	33340	Peonidin-3-glucoside chloride
88	1086	2,2,2-Trifluoroethanol	192	33341	Malvidin-3,5-diglucoside chloride
89	1090	2,2-Dimethylpentane	193	34501	2-Hydroxypropyl-beta-cyclodextrin
90	1143	1,3-Butanediol	194	34550	1,8-Bis(trimethylammonium)octane dibromide
91	1264	alpha-Aminotoluene	195	34551	1,10-Bis(trimethylammonium)decane dibromide
92	1292	Helium	196	34552	1,12-Bis(trimethylammonium)dodecane dibromide
93	1293	Neon	197	36721	o-Sulfanilic acid
94	1594	Pyrene	198	37864	2-Hydroxypropyl-alpha-cyclodextrin
95	1642	1,4-Dihydroxybenzene	199	40775	DL-m-Tyrosine
96	1645	1,2,3-Trihydroxybenzene	200	40777	DL-o-Tyrosine
97	2186	Diisopropanolamine	201	40779	D,L-beta-Aminoisobutyric acid
98	2187	Methyldiethanolamine	202	43996	m-Sulfanilic acid
99	2245	Phosphoric acid	203	46014	p-Phenolsulfonic acid
100	2501	1,2-Benzenediamine	204	49211	beta-Cyclodextrin, sulfated sodium salt
101	2506	3-Methoxyphenol	205	51976	Lithium acetylacetonate
102	2542	Perylene	206	54011	N-Methylphenothiazine
103	2945	3-Hydroxyaniline	207	54491	L-Histidine methyl ester dihydrochloride
104	2994	DL-Tyrosine	208	61801	Tetrasodium tetraphenylporphyrintetrasulfonate
Solvents					
1	3	Acetonitrile	27	161	Toluene
2	4	Acetone	28	174	Water
3	11	Ethanol	29	250	Cyclohexanone
4	12	Diethyl ether	30	282	1,2-Propanediol
5	21	Ethyl acetate	31	284	N-Methyl-2-pyrrolidone
6	25	Ethylbenzene	32	297	Hexafluorobenzene
7	26	Bromobenzene	33	367	2,3-Dimethylbutane
8	27	Chlorobenzene	34	430	Methyl isopropyl ketone
9	30	Nitrobenzene	35	451	Carbonic acid dimethyl ester

Table D.1 continued.

Cons. No.	DDB No.	Name	Cons. No.	DDB No.	Name
10	31	Benzene	36	516	Hexadecane
11	39	1-Butanol	37	887	Deuterium oxide
12	40	2-Butanone	38	982	Perdeuteromethanol
13	46	Butyl chloride	39	1090	2,2-Dimethylpentane
14	47	Chloroform	40	3410	1,3,5-Triisopropylbenzene
15	50	Cyclohexane	41	4331	Hexamethyltetracosane
16	60	Decane	42	16447	1-Ethyl-3-methylimidazolium bis(trifluoromethylsulfonyl)imide
17	72	N,N-Dimethylformamide	43	16583	1-Butyl-3-methylimidazolium bis(trifluoromethylsulfonyl)imide
18	78	Dodecane	44	16810	1-Octyl-3-methylimidazolium tetrafluoroborate
19	80	Butyl acetate	45	18162	1-Hexyl-3-methylimidazolium bis(trifluoromethylsulfonyl)imide
20	89	Hexane	46	18174	1-Hexyl-3-methylimidazolium tetrafluoroborate
21	91	Heptane	47	18642	1-Ethyl-3-methylimidazolium bis(pentafluoroethylsulfonyl)imide
22	97	2,2,4-Trimethylpentane	48	18988	1-Ethyl-3-methylimidazolium trifluoromethylsulfonate
23	110	Methanol	49	20138	1-Butyl-3-methylimidazolium dicyanamide
24	112	3-Methylpentane	50	22417	1-Ethyl-3-methylimidazolium trifluoroacetate
25	140	1-Propanol	51	22674	1-Butyl-3-methylpyridinium tetrafluoroborate
26	157	Tetrachloromethane			

Table D.2: Latent variables u_i of the solutes for both hybrid MCMs, for the data set of the reduced database.

		MCM-Boosting		MCM-Whisky	
i	**Name**	u_{i1}	u_{i2}	u_{i1}	u_{i2}
1	Acetonitrile	0.0259	-0.3137	1.1140	1.0072
2	Acetone	-0.0383	-0.6594	0.8325	1.0875
3	Ethanol	-0.1208	-0.2232	1.2558	0.6697
4	Ethyl acetate	0.0183	-1.0212	1.2218	0.3069
5	Benzyl alcohol	-0.0248	0.0109	0.9067	-0.2790
6	Ethylbenzene	0.0853	-0.3729	1.0475	-0.0265
7	Chlorobenzene	0.0830	-0.6592	1.0640	-0.0972
8	Benzene	0.0225	-0.7974	1.0589	0.6732
9	1-Butanol	0.0652	0.5200	0.8152	-0.3123
10	Butyl chloride	-0.1005	-1.3470	0.9289	1.0876
11	3-Methylphenol	0.0168	0.0857	0.9696	-0.3696
12	Cyclohexane	-0.0247	-0.1559	0.9850	0.0256
13	Dodecane	-0.1310	0.4674	0.7540	-0.2834
14	Benzaldehyde	-0.0026	-0.3415	0.9643	0.0233

Table D.2 continued.

		MCM-Boosting		MCM-Whisky	
i	**Name**	u_{i1}	u_{i2}	u_{i1}	u_{i2}
15	Butyl acetate	0.0583	-0.5567	1.0636	-0.1444
16	Acetic acid	0.1207	0.3451	1.0522	0.0817
17	Hexane	-0.0562	0.4757	1.0284	0.1294
18	Heptane	-0.1007	-0.3165	1.0001	-0.0528
19	Methyl iodide	0.0453	-1.8181	1.1878	0.5537
20	Methanol	-0.0717	0.6005	1.2734	0.8127
21	Naphthalene	0.1071	-0.6001	1.0165	-0.0713
22	Phenol	0.1714	-0.1511	1.1026	-0.1210
23	1-Propanol	0.0338	0.4886	1.0087	-0.1563
24	Propionic acid	0.0481	0.1907	1.0585	0.0166
25	Tetrachloromethane	0.0657	-0.7153	0.9538	-0.2629
26	Toluene	0.0294	-0.4151	1.0385	0.1623
27	Water	0.0160	1.0622	1.2625	1.2361
28	Glycerol	-0.0547	0.2778	0.9945	-0.4034
29	Butyric acid	0.0493	0.1375	0.9714	-0.0868
30	Hexafluorobenzene	0.0554	-1.2698	1.0224	-0.0083
31	2-Methyl-2,4-pentanediol	0.1493	-0.1078	0.7891	-0.4680
32	Acetophenone	0.0366	-0.2193	0.8595	-0.0045
33	Methyl isopropyl ketone	-0.0220	-0.3874	1.0480	0.2781
34	Hexadecane	-0.0178	0.6118	0.6210	-0.4563
35	p-Chlorotoluene	0.1212	-0.5070	1.0570	-0.1735
36	1,2-Dihydroxybenzene	0.0735	0.3086	0.8749	-0.5178
37	p-Chlorophenol	0.0550	-0.0153	0.9082	-0.4891
38	1,3-Dihydroxybenzene	0.1110	0.5870	1.0430	-0.8759
39	Carbon dioxide	0.0024	-2.1624	1.0677	2.6978
40	Pyrene	0.0411	-0.3815	0.7835	-0.4026
41	1,4-Dihydroxybenzene	-0.0929	0.7952	1.0314	-0.9901
42	1,2,3-Trihydroxybenzene	0.0466	0.5331	0.8596	-0.8712

Table D.2 continued.

		MCM-Boosting		MCM-Whisky	
i	**Name**	u_{i1}	u_{i2}	u_{i1}	u_{i2}
43	Perylene	-0.0153	-0.2712	0.6542	-0.6323
44	3-Hydroxyaniline	0.0221	0.5058	0.9982	-0.6519
45	Di-tert-butylsulfide	-0.0455	-0.5260	0.8416	0.0409

Table D.3: Latent variables v_j of the solvents for both hybrid MCMs, for the data set of the reduced database.

		MCM-Boosting		MCM-Whisky	
j	**Name**	v_{j1}	v_{j2}	v_{j1}	v_{j2}
1	Acetonitrile	0.0808	0.1552	1.0329	0.3479
2	Acetone	-0.0143	0.3749	1.2012	0.0947
3	Ethanol	0.0235	0.4952	-0.0895	0.4689
4	Ethyl acetate	0.0245	-0.4662	0.5537	0.1355
5	Benzene	0.0041	0.3100	0.7094	0.2268
6	1-Butanol	-0.0171	0.6077	-0.7692	0.4649
7	Butyl chloride	-0.0616	0.0769	1.0076	0.0935
8	Chloroform	-0.1435	0.0860	0.7535	0.2162
9	Cyclohexane	-0.0421	0.4390	0.3997	0.2822
10	Dodecane	0.1043	-0.6007	0.1037	0.2835
11	Butyl acetate	-0.0016	0.3710	0.5711	0.2338
12	Hexane	-0.0297	0.0871	1.2860	0.0933
13	Heptane	-0.0449	0.1479	1.1070	0.1109
14	Methanol	-0.0325	0.3837	0.5793	0.4010
15	1-Propanol	0.0604	0.5545	-0.6209	0.5466
16	Tetrachloromethane	0.0471	-0.0292	0.2200	0.2178
17	Toluene	-0.0317	0.0448	0.7320	0.1470
18	Water	0.0019	0.1685	0.0015	0.2546
19	1,2-Propanediol	-0.0228	0.4908	-3.1392	0.5874

Table D.3 continued.

		MCM-Boosting		MCM-Whisky	
j	**Name**	v_{j1}	v_{j2}	v_{j1}	v_{j2}
20	N-Methyl-2-pyrrolidone	-0.0056	0.2545	-0.4168	0.3778
21	Hexafluorobenzene	-0.0188	-0.0455	0.3585	0.1540
22	Methyl isopropyl ketone	-0.0201	0.3742	0.7499	0.1693
23	Hexadecane	-0.0470	-1.1143	-0.3064	1.1473

D.6 Stan Code

In the following, the Stan codes for training all MCMs used in this chapter are provided: the data-driven MCM, MCM-Boosting, and MCM-Whisky. For MCM-Whisky, the codes of the two training steps, distillation and maturation, are given individually. An executable form of this code is included for download in the form of .stan files in Ref. [83]. To run the code, users will need to install an interface of their choice from the project's homepage (`https://mc-stan.org/users/interfaces/`). For further information, Stan's excellent documentation is referred to: `https://mc-stan.org/users/documentation/`.

Furthermore, Ref. [83] provides a wrapper code for each MCM, i.e., a MATLAB script that reads the training data from a .csv file, applies the developed MCMs for the prediction of the full matrix, and exports the result to a .csv file.

D.6.1 Data-Driven MCM

```
1 data {
2   int<lower=0> I; // number of solutes
3   int<lower=0> J; // number of solvents
4   int<lower=0> K; // number of latent dimensions
5   real ln_D[I,J]; // matrix of logarithmic diffusion
      coefficients
6   real<lower=0> sigma_0; // prior standard deviation
7   real<lower=0> lambda; // likelihood scale
8 }
```

```
9
10 parameters {
11   vector[K] u[I]; // solute feature vectors
12   vector[K] v[J]; // solvent feature vectors
13 }
14
15 model {
16   // prior: draw feature vectors for all solutes and
        solvents:
17   for (i in 1:I)
18     u[i] ~ normal(0,sigma_0);
19   for (j in 1:J)
20     v[j] ~ normal(0,sigma_0);
21   // likelihood: model the probability of ln_D as a normal
        distribution
22   // around the dot product of the feature vectors:
23   for (i in 1:I) {
24     for (j in 1:J) {
25       if (ln_D[i,j] != -99) { // train to available data
            only
26         ln_D[i,j] ~ normal(u[i]' * v[j], lambda);
27       }
28     }
29   }
30 }
```

D.6.2 MCM-Boosting

```
1 data {
2   int<lower=0> I; // number of solutes
3   int<lower=0> J; // number of solvents
4   int<lower=0> K; // number of latent dimensions
5   real R[I,J]; // matrix of residuals of logarithmic
       diffusion coefficients
6   real<lower=0> sigma_0; // prior standard deviation
7   real<lower=0> lambda; // likelihood scale
8 }
9
```

```
parameters {
  vector[K] u[I]; // solute feature vectors
  vector[K] v[J]; // solvent feature vectors
}

model {
  // prior: draw feature vectors for all solutes and
     solvents:
  for (i in 1:I)
    u[i] ~ normal(0,sigma_0);
  for (j in 1:J)
    v[j] ~ normal(0,sigma_0);
  // likelihood: model the probability of R as a normal
     distribution around the dot product of the feature
     vectors:
  for (i in 1:I) {
    for (j in 1:J) {
      if (R[i,j] != -99) { // train to available data only
        R[i,j] ~ normal(u[i]' * v[j], lambda);
      }
    }
  }
}
```

D.6.3 MCM-Whisky: Distillation

```
data {
  int<lower=0> I; // number of solutes
  int<lower=0> J; // number of solvents
  int<lower=0> K; // number of latent dimensions
  real ln_D[I,J]; // matrix of logarithmic diffusion
     coefficients
  real<lower=0> sigma_0; // prior standard deviation
  real<lower=0> lambda; // likelihood scale
}

parameters {
  vector[K] u[I]; // solute feature vectors
```

```
12   vector[K] v[J]; // solvent feature vectors
13 }
14
15 model {
16   // prior: draw feature vectors for all solutes and
        solvents:
17   for (i in 1:I)
18     u[i] ~ normal(0,sigma_0);
19   for (j in 1:J)
20     v[j] ~ normal(0,sigma_0);
21   // likelihood: model the probability of ln_D as a normal
        distribution around the dot product of the feature
        vectors:
22   for (i in 1:I) {
23     for (j in 1:J) {
24       if (ln_D[i,j] != -99) { // train to available data
            only
25         ln_D[i,j] ~ cauchy(u[i]' * v[j], lambda);
26       }
27     }
28   }
29 }
```

D.6.4 MCM-Whisky: Maturation

```
1 data {
2   int<lower=0> I; // number of solutes
3   int<lower=0> J; // number of solvents
4   int<lower=0> K; // number of latent dimensions
5   real ln_D[I,J]; // matrix of logarithmic diffusion
       coefficients
6   real<lower=0> lambda; // likelihood scale
7   vector<lower=0>[K] sigma_0_u[I]; // Prior standard
       deviation (Solutes)
8   vector<lower=0>[K] sigma_0_v[J]; // Prior standard
       deviation (Solvents)
9   vector[K] mu_0_u[I]; // prior mean (Solutes)
10  vector[K] mu_0_v[J]; // prior mean (Solvents)
```

```
}

parameters {
  vector[K] u[I]; // solute feature vectors
  vector[K] v[J]; // solvent feature vectors
}

model {
  // prior: draw feature vectors for all solutes and
     solvents:
  for (i in 1:I)
    u[i] ~ normal(mu_0_u[i],sigma_0_u[i]);
  for (j in 1:J)
    v[j] ~ normal(mu_0_v[j],sigma_0_v[j]);
  // likelihood: model the probability of ln_D as a normal
     distribution around the dot product of the feature
     vectors:
  for (i in 1:I) {
    for (j in 1:J) {
      if (ln_D[i,j] != -99) {//available data only
        ln_D[i,j] ~ normal(u[i]' * v[j], lambda);
      }
    }
  }
}
```

E Supporting Information for Chapter 5.1

E.1 UNIFAC Model

E.1.1 UNIFAC Equations

For predicting the logarithmic activity coefficient $\ln \gamma_i$ of a component i in a mixture, the UNIFAC model considers the sum of an entropic contribution, called combinatorial part, $\ln \gamma_i^{\mathrm{C}}$ and an energetic contribution, called residual part, $\ln \gamma_i^{\mathrm{R}}$ [41]:

$$\ln \gamma_i = \ln \gamma_i^{\mathrm{C}} + \ln \gamma_i^{\mathrm{R}} \tag{E.1}$$

The combinatorial part $\ln \gamma_i^{\mathrm{C}}$ is thereby calculated by:

$$\ln \gamma_i^{\mathrm{C}} = 1 - V_i + \ln V_i - \frac{z}{2} q_i \left(1 - \frac{V_i}{F_i} + \ln \frac{V_i}{F_i}\right) \tag{E.2}$$

with

$$V_i = \frac{r_i}{\sum_j r_j x_j} \tag{E.3}$$

$$F_i = \frac{q_i}{\sum_j q_j x_j} \tag{E.4}$$

where r_i and q_i are the relative Van der Waals volume and surface area of component i, respectively, x_i is the mole fraction of component i in the mixture, and z is the coordination number, which is set to $z = 10$ in basically all cases and was also used here. Eqs. (E.1) - (E.4) are identical to the equations used in the UNIQUAC model [116]; the difference between UNIQUAC and UNIFAC is that UNIQUAC is based on *component-specific* parameters, whereas they are derived from *group-specific* parameters in UNIFAC. Specifically, in UNIFAC, the relative Van der Waals volume r_i and surface area q_i of the component i are calculated from the group volume and group surface parameters, R_k and Q_k, respectively, which are tabulated for multiple structural groups k [12, 119–123],

as follows:

$$r_i = \sum_k \nu_k^{(i)} R_k \tag{E.5}$$

$$q_i = \sum_k \nu_k^{(i)} Q_k \tag{E.6}$$

where $\nu_k^{(i)}$ denotes the frequency of group k in one molecule of component i.

The residual part $\ln \gamma_i^{\mathrm{R}}$ of UNIFAC is calculated by:

$$\ln \gamma_i^{\mathrm{R}} = \sum_k \nu_k^{(i)} \left(\ln \Gamma_k - \ln \Gamma_k^{(i)} \right) \tag{E.7}$$

where Γ_k is the group activity coefficient of group k in the mixture and $\Gamma_k^{(i)}$ is the group activity coefficient of group k in the pure component i. Both Γ_k and $\Gamma_k^{(i)}$ are calculated similar to the residual part in the UNIQUAC model by:

$$\ln \Gamma_k = Q_k \left(1 - \ln \left(\sum_m \Theta_m \Psi_{mk} \right) - \sum_m \frac{\Theta_m \Psi_{km}}{\sum_n \Theta_n \Psi_{nm}} \right) \tag{E.8}$$

where Θ_m is the surface fraction of group m in the mixture:

$$\Theta_m = \frac{Q_m X_m}{\sum_n Q_n X_n} \tag{E.9}$$

and X_m is the group mole fraction of group m, which is related to the mole fractions x_j of components j:

$$X_m = \frac{\sum_j \nu_m^{(j)} x_j}{\sum_j \sum_n \nu_n^{(j)} x_j} \tag{E.10}$$

The parameters Ψ_{nm} and Ψ_{mn} in Eq. (E.8) contain the group-interaction parameters of UNIFAC, a_{nm} and a_{mn}, between the groups m and n:

$$\Psi_{nm} = \exp\left(-\frac{a_{nm}}{T} \right); \qquad \Psi_{mn} = \exp\left(-\frac{a_{mn}}{T} \right) \tag{E.11}$$

E.1.2 UNIFAC Group-Interaction Parameters

In Fig. E.1, the current availability of group-interaction parameters of the public UNIFAC [12] and the commercial UNIFAC-TUC [127] is indicated.

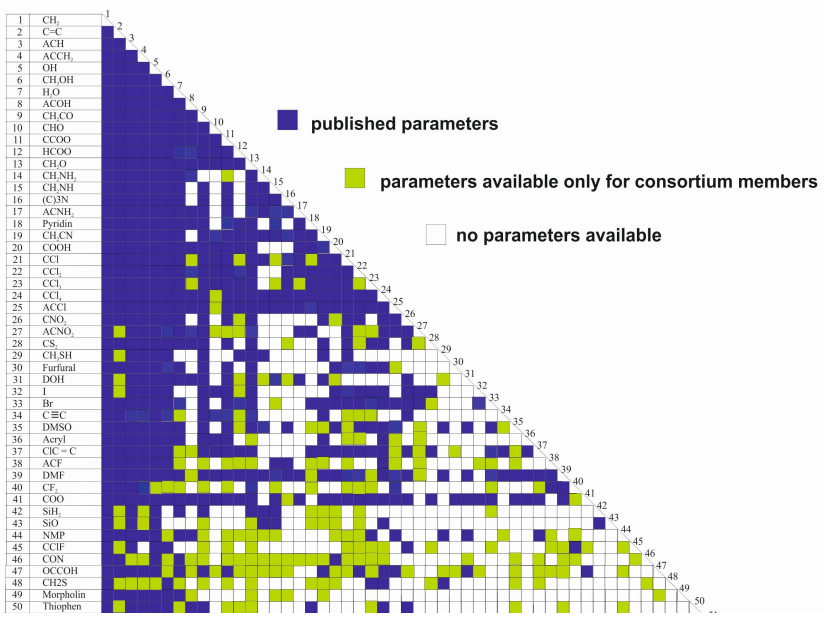


Figure E.1: Matrix representing the availability of group-interaction parameters of the public UNIFAC [12] (blue) and the commercial UNIFAC-TUC [127] (green) up to main group 50. White cells: no parameters available.

E.2 Model Details

E.2.1 Bayesian Matrix Completion

The model of the present chapter is similar to the one recently introduced in Ref. [84], in which a matrix completion method (MCM) from machine learning has been combined with the UNIQUAC model [116, 117]. In contrast to Ref. [84], the group-interaction parameters among *structural groups* m and n (and not components), specifically between main groups of UNIFAC, are predicted here. Fig. E.2 shows an overview of the proposed UNIFAC-MCM model as well as of the training and evaluation procedure.

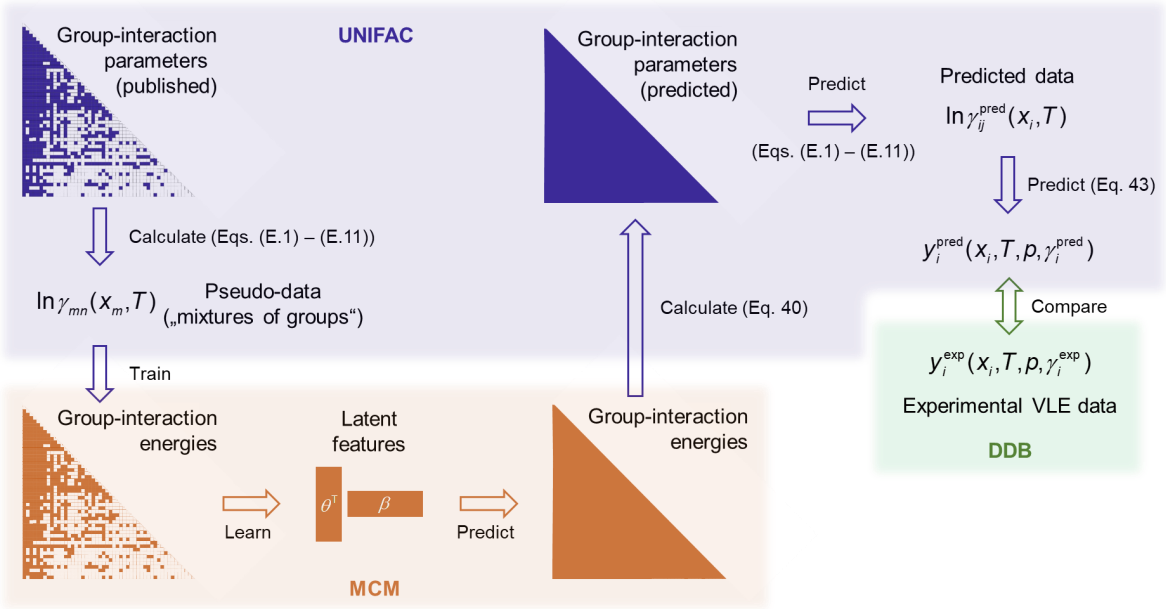


Figure E.2: Scheme representing the training and evaluation of UNIFAC-MCM. Besides based on the vapor-phase composition y, the results were also evaluated based on deviations in the temperature T and the pressure p from the experimental vapor-liquid equilibrium (VLE) data from the Dortmund Data Bank (DDB).

The model has been trained on pseudo-data for logarithmic activity coefficients $\ln\gamma_{mn}$ in hypothetical binary mixtures of groups, m and n, which have been generated with the UNIFAC model using the current public parameterization [12] as described in Chapter 5.1. Note that although these pseudo-data were generated based on an *inconsistent* set of group-interaction parameters, the pseudo-data themselves are not inconsistent, because very similar activity coefficients can be obtained by different combinations of group-interaction parameters. This makes the values of the group-interaction parameters less informative, whereas the generated pseudo-data contain the structure that is recovered by the MCM during the training.

Thereby, a Bayesian approach to matrix completion (cf. Chapter 2) has been employed, which consists of multiple steps as described in the following.

First, a generative probabilistic model for $\ln \gamma_{mn}$ as a nonlinear function f of the groups m and n, the temperature T, and the mole fraction x_m of group m in the hypothetical mixture has been specified. This function is basically defined by the UNIFAC equations, cf. Eqs. (E.1) - (E.11), the correlation of the group-interaction parameters a_{mn} and a_{nm} via the group-interaction energies U_{mn}, U_{mm}, and U_{nn}, cf. Eqs. (40) and (41), and an embedded matrix factorization for the unlike group-interaction energies U_{mn} between the groups, cf. Eq. (42). The function furthermore considers the following parameters:

- group-specific parameters considered in the UNIFAC model, specifically the group volume parameters R_m and R_n and the group surface parameters Q_m and Q_n, which were adopted from the latest public parameter table of UNIFAC, cf. Table E.1;
- initially unknown (latent) feature vectors $\boldsymbol{\theta}_m$, $\boldsymbol{\theta}_n$, $\boldsymbol{\beta}_m$, and $\boldsymbol{\beta}_n$ of the groups, which are used for modeling the unlike group-interaction energies U_{mn} between the groups, as well as the like group-specific group-interaction energies U_{mm} and U_{nn}.

The length K of the feature vectors, which controls the number of features that are considered for each group, is in principle a hyperparameter of the model, which can be adjusted during model selection. However, in this chapter, a comprehensive hyperparameter screening has not been carried out, and the hyperparameters from Ref. [84], including the setting of $K = 3$, have simply been adopted.

Table E.1: UNIFAC main groups m considered in the present chapter and the respective group volume and group surface parameters, R_m and Q_m, used. Some main groups include multiple subgroups, such that R_m and Q_m could have been chosen differently, whereby, however, no large impact is expected; in such cases, usually one of the "intermediate" subgroups was chosen randomly here (e.g., "CH2").

m	R_m	Q_m	m	R_m	Q_m
1	0.6744	0.54	26	1.7818	1.56
2	1.1167	0.867	27	1.4199	1.104
3	0.5313	0.4	28	2.057	1.65
4	1.0396	0.66	29	1.651	1.368
5	1	1.2	30	3.168	2.484
6	1.4311	1.432	31	2.4088	2.248
7	0.92	1.4	32	1.264	0.992
8	0.8952	0.68	33	0.9492	0.832
9	1.4457	1.18	34	1.0613	0.784
10	0.998	0.948	35	2.8266	2.472
11	1.6764	1.42	36	2.3144	2.052
12	1.242	1.188	37	0.791	0.724
13	0.9183	0.78	38	0.6948	0.524
14	1.3692	1.236	39	3.0856	2.736
15	1.207	0.936	40	1.0105	0.92
16	0.9597	0.632	41	1.38	1.2
17	1.06	0.816	42	1.4443	1.0063
18	2.8332	1.833	43	1.303	0.7639
19	1.6434	1.416	44	3.981	3.2
20	1.3013	1.224	45	2.2287	1.916
21	1.238	0.952	46	1.9637	1.488
22	2.0606	1.684	47	1.8952	1.592
23	2.6401	2.184	48	1.3863	1.06
24	3.39	2.91	49	3.474	2.796
25	1.1562	0.844	50	2.6908	1.86

$\boldsymbol{\theta}_m$, $\boldsymbol{\theta}_n$, $\boldsymbol{\beta}_m$, $\boldsymbol{\beta}_n$, U_{mm}, and U_{nn} constitute the parameters of the model that were inferred during the training. For the training, the generative model defines a probability distribution over all used pseudo-data for $\ln \gamma_{mn}$ by specifying a stochastic process for generating hypothetical data for $\ln \gamma_{mn}$ conditioned on $\boldsymbol{\theta}_m$, $\boldsymbol{\theta}_n$, $\boldsymbol{\beta}_m$, $\boldsymbol{\beta}_n$, U_{mm}, and U_{nn}, which are initially unknown, R_m, R_n, Q_m, and Q_n, which were adopted from Refs. [41, 124–126], and the temperature and the mole fraction of m in the mixture. The generative process therefore draws $\boldsymbol{\theta}_m$, $\boldsymbol{\theta}_n$, $\boldsymbol{\beta}_m$, $\boldsymbol{\beta}_n$, U_{mm}, and U_{nn} from a normal prior distribution with zero mean and a standard deviation of one. The type of distribution used as prior as well as the mean and the standard deviation are also hyperparameters of the model, but were, as K, also set as in Ref. [84]. Then, the generative process models the probability of the training data $\ln \gamma_{mn}$ as a Cauchy likelihood distribution with scale $\lambda = 0.2$ centered around the outcome of the function f with the $\boldsymbol{\theta}_m$, $\boldsymbol{\theta}_n$, $\boldsymbol{\beta}_m$, $\boldsymbol{\beta}_n$, U_{mm}, and U_{nn} drawn from the prior and the fixed parameters and conditions. Again, the type of distribution used as likelihood as well as the scale are hyperparameters, which were set as in Ref. [84]. The likelihood can be written as follows:

$$\ln \gamma_{mn}(T, x_m) = \text{Cauchy}(f(T, x_m, R_m, R_n, Q_m, Q_n, \boldsymbol{\theta}_m, \boldsymbol{\theta}_n, \boldsymbol{\beta}_m, \boldsymbol{\beta}_n, U_{mm}, U_{nn}), \lambda) + \epsilon_{mn} \tag{E.12}$$

where the function f includes the UNIFAC equations, Eqs. (E.1) - (E.11), as well as Eqs. (40) - (42) and ϵ_{mn} captures the deviations between the model results and the pseudo-data $\ln \gamma_{mn}(T, x_m)$ for training the model. In the next step, the parameters that were to be learned, i.e., $\boldsymbol{\theta}_m$, $\boldsymbol{\theta}_n$, $\boldsymbol{\beta}_m$, $\boldsymbol{\beta}_n$, U_{mm}, and U_{nn}, were concurrently inferred for all groups m based on the set of training data, which requires the inversion of the generative model. Since full Bayesian inference is intractable except for very simple cases, Gaussian mean-field variational inference [15, 18, 19] was used for this purpose. Simply put, this procedure can be understood as a comparison of the generated hypothetical $\ln \gamma_{mn}$ to the training data, i.e., the pseudo-data for $\ln \gamma_{mn}$ as obtained with UNIFAC using the latest public parameterization, to subsequently adjust the initially unknown parameters. This results in the posterior, which constitutes a probability distribution for all inferred parameters.

Finally, the means of the approximated posterior distributions over $\boldsymbol{\theta}_m$, $\boldsymbol{\theta}_n$, $\boldsymbol{\beta}_m$, $\boldsymbol{\beta}_n$, U_{mm}, and U_{nn} were used to predict the group-interaction parameters a_{mn} and a_{nm} for all possible combinations of groups according to Eqs. (40) - (42). The predicted a_{mn} and a_{nm} were, in turn, used for predicting the activity coefficients of *components* $\ln \gamma_i$ in binary mixtures with Eqs. (E.1) - (E.11), which were finally used for predicting vapor-liquid equilibrium (VLE) phase diagrams. This approach thereby basically changes Eq. (E.11) to:

$$\Psi_{nm} = \exp\left(-\frac{\boldsymbol{\theta}_n \cdot \boldsymbol{\beta}_m + \boldsymbol{\theta}_m \cdot \boldsymbol{\beta}_n - U_{mm}}{T}\right); \qquad \Psi_{mn} = \exp\left(-\frac{\boldsymbol{\theta}_n \cdot \boldsymbol{\beta}_m + \boldsymbol{\theta}_m \cdot \boldsymbol{\beta}_n - U_{nn}}{T}\right) \tag{E.13}$$

The predicted VLE phase diagrams were compared to experimental data from the Dortmund Data Bank (DDB) [137].

For performing the task of Bayesian inference, the Stan framework [20] was used.

E.2.2 Scope of UNIFAC-MCM

Since UNIFAC-MCM yields a complete set of group-interaction parameters for the first 50 main groups of UNIFAC, the approach allows modeling any binary and multicomponent mixture whose components can be built from these groups. The scope of the new approach is thereby much larger than can be demonstrated here, simply due to missing experimental data for a more comprehensive assessment. This is also indicated in Fig. E.3, which shows the number of binary systems from the data set for which VLE data are available in the DDB [137] and which contain the respective combination of UNIFAC main groups.

While there are several group-interaction parameters that are required for modeling a large number of binary systems for the data set (dark-colored cells in Fig. E.3), approximately 80% of all possible main group combinations are not represented in the data set (white cells in Fig. E.3). The lack of experimental data inevitably prevents the parameterization of UNIFAC, both in its public and commercial versions, in the ordinary way, as only those parameters can be fitted for which respective training data are available. With UNIFAC-MCM, on the other hand, this problem is solved; UNIFAC-MCM yields predictions also for the 80% of group-interaction parameters from Fig. E.3 for which classical UNIFAC versions cannot achieve this based on the studied data set.

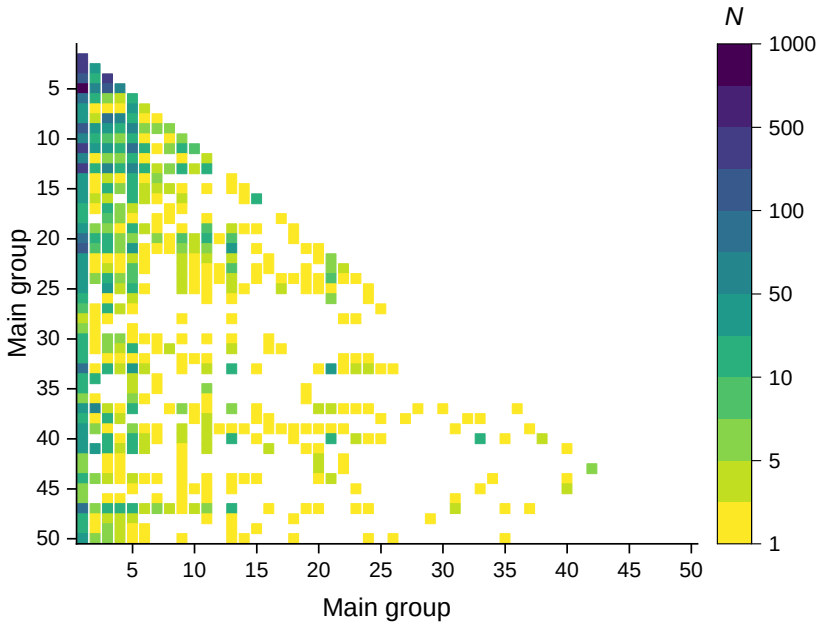


Figure E.3: Heatmap showing the number N of binary systems for which VLE data are available in the DDB and which contain the respective combination of UNIFAC main groups. White cells indicate that no VLE data are available for the given combination of groups.

E.3 Additional Results

In Fig. E.4, the results of UNIFAC-MCM for the prediction of the VLE data are plotted in histogram representations, which show the number of binary systems that are predicted with a defined relative deviation from the experimental mole fraction of the low-boiling component in the vapor phase Δy. In the left panel, the results of UNIFAC-MCM on the complete horizon are shown. In the middle panel, the results for those systems from the complete horizon are shown that can not be modeled by the public UNIFAC version, but by the commercial UNIFAC-TUC; here, the UNIFAC-MCM predictions are compared to those of UNIFAC-TUC. In the right panel, the results for those systems from the complete horizon are shown that can only be modeled by UNIFAC-MCM.

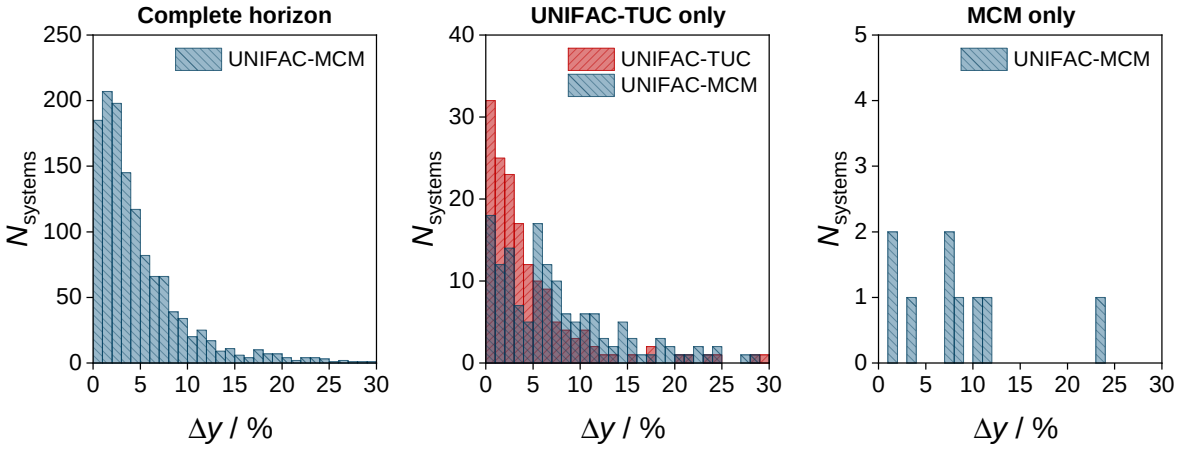


Figure E.4: Histogram representations of number of systems that are predicted by UNIFAC-MCM with a defined relative deviation from the experimental vapor mole fractions of the low-boiling components Δy. Left: for the complete horizon (2,246 systems). Middle: for those systems that can not be predicted with public UNIFAC (169 systems). Right: for those systems that can not be predicted with UNIFAC-TUC (9 systems).

F Supporting Information for Chapter 5.2.1

F.1 UNIFAC Parameterization

Fig. F.1 (a) visualizes which pair-interaction parameters a_{mn} are already reported in UNIFAC 1.0 and which a_{mn} can additionally be fitted to the considered database. The heatmap in Fig. F.1 (b) indicates the number of experimental data points from the DDB for which a specific a_{mn} is relevant. The figure reveals an extreme heterogeneity, e.g., while 109 a_{mn} (7.6% of the matrix) is required for at least 1,000 data points, 476 (33%) are not represented in any available data point.

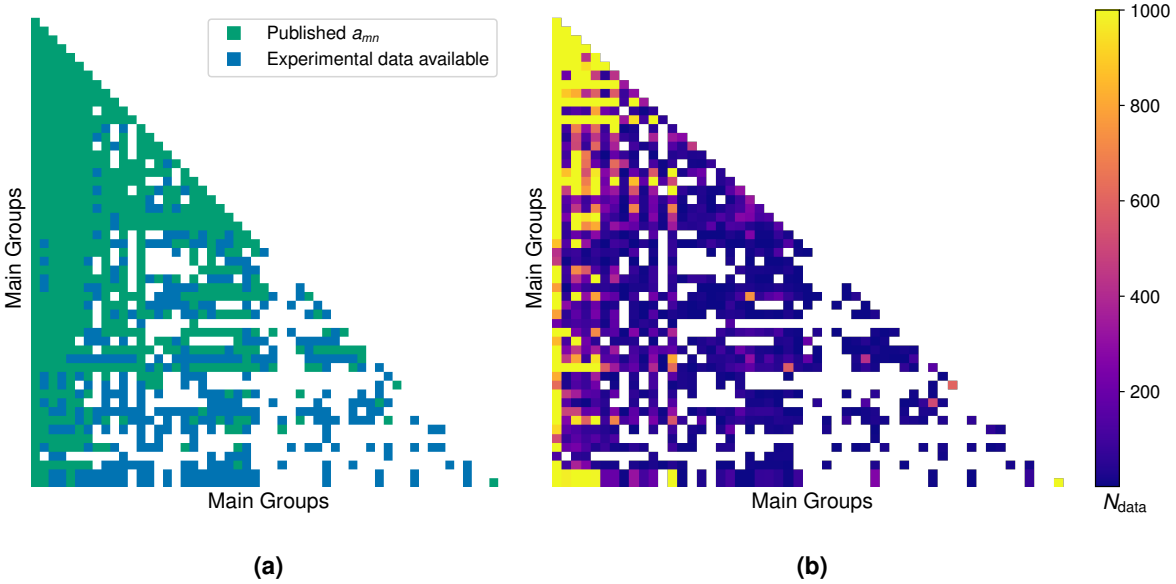


Figure F.1: (a) Representation of the published UNIFAC pair-interaction parameters a_{mn} [12] (green) and the ones that could additionally be fitted using the experimental data from the DDB [38] (blue). (b) Heatmap of number of experimental data points from the DDB requiring specific a_{mn}.

Fig. F.2 is an extension of Fig. F.1 (a), additionally including the interaction parameters available for members of the UNIFAC-Consortium. Note that more than the 54 considered main groups are defined for this UNIFAC variant, which are omitted here.

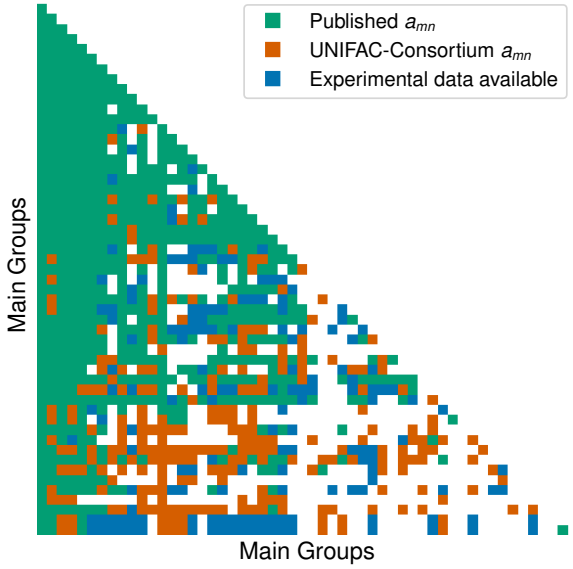


Figure F.2: Matrix of existing UNIFAC parameters (a_{mn}) of the public UNIFAC 1.0 model [12] (green), supplemented by those of the commercial UNIFAC-Consortium variant [141] (orange). Furthermore, group combinations are marked, for which data are available, but no parameters have yet been fitted (blue).

Although the UNIFAC-Consortium model has a substantially increased scope compared to UNIFAC 1.0, Fig. F.2 still reveals significant gaps in the interaction parameter matrix, which can be mainly attributed to the lack of available experimental training data. This underlines the importance of methods like UNIFAC 2.0, which can extrapolate these missing interaction parameters. Since the parameter tables of the UNIFAC-Consortium model are not disclosed, an evaluation and comparison of its prediction accuracy could not be conducted here.

F.2 Prediction Accuracy for Selected Binary Mixtures

Table F.1 lists the 20 binary mixtures with the greatest improvement in mean squared error (MSE) achieved by UNIFAC 2.0 compared to UNIFAC 1.0. Most of these mixtures involve either a methoxy group paired with a silane group (main groups 13 ("CH2O") and 42 ("SIH2")) or water paired with chlorinated aromatic components (main groups 7 ("H2O") and 25 ("ACCL")). This highlights that the corresponding pair-interaction parameters in UNIFAC 1.0 are poorly fitted, whereas UNIFAC 2.0 provides a significantly more accurate representation.

Table F.1: Mean absolute error (MAE) and mean squared error (MSE) of predictions using UNIFAC 1.0 and UNIFAC 2.0 for binary mixtures showing the largest improvement in MSE.

Component i	Component j	$\mathrm{MAE}_{\mathrm{mix}}^{2.0}$	$\mathrm{MSE}_{\mathrm{mix}}^{2.0}$	$\mathrm{MAE}_{\mathrm{mix}}^{1.0}$	$\mathrm{MSE}_{\mathrm{mix}}^{1.0}$
Octane	Tetramethoxysilane	0.12	0.03	8.27	137.06
Heptane	Tetramethoxysilane	0.06	0.01	7.42	110.19
Water	1,2-Dimethoxy-3,4,5,6-tetrachlorobenzene	0.08	0.01	7.19	103.46
Ethylcyclohexane	Tetramethoxysilane	0.13	0.03	7.21	104.22
Hexane	Tetramethoxysilane	0.00	0.00	6.56	86.24
Methylcyclohexane	Tetramethoxysilane	0.06	0.01	6.37	81.22
Cyclohexane	Tetramethoxysilane	0.05	0.01	5.57	62.02
Water	1,2-Dimethoxy-3,4,5-trichlorobenzene	0.23	0.11	5.72	65.33
Water	1,2,4,5-Tetrachloro-3-methyl-6-isopropylbenzene	0.61	0.75	5.90	69.73
Water	2,3,4-Trichloroanisole	0.03	0.00	5.20	54.11
Water	2,4,6-Trichloroanisole	0.02	0.00	5.20	54.04
Water	2,4,6-Trichlorophenol	0.28	0.20	5.40	59.65
Water	2,3,4,5-Tetrachloroanisole	1.09	2.40	5.70	64.91

Table F.1 continued.

Component i	Component j	$MAE_{mix}^{2.0}$	$MSE_{mix}^{2.0}$	$MAE_{mix}^{1.0}$	$MSE_{mix}^{1.0}$
Water	1,2,3-Trichlorobenzene	0.17	0.06	4.60	42.50
Water	2,3,5,6-Tetrachloroanisole	1.19	2.85	5.60	62.69
Water	1,2,4-Trichlorobenzene	0.30	0.21	4.46	39.86
Phenol	Tetrachloromethane	0.25	0.15	4.02	70.14
Benzene	Tetramethoxysilane	0.18	0.07	3.90	30.42
Water	2,4-Dichlorophenol	0.12	0.07	3.75	28.60
Water	2,6-Dichloroanisole	0.04	0.00	3.59	25.83

F.3 Sensitivity of the Selected Hyperparameters

Fig. F.3 shows the performance of UNIFAC 2.0 with varied latent dimension K and scale parameter of the Cauchy likelihood λ on the test set in the extrapolation study considering unstudied components, cf. Fig. 41. In both cases, UNIFAC 2.0 is rather robust towards changes in the hyperparameters, with only very small values of K or λ showing a significant deterioration in prediction accuracy. This underscores the choice of setting the hyperparameters to $K = 8$ and $\lambda = 0.4$. However, other suitable values are also possible.

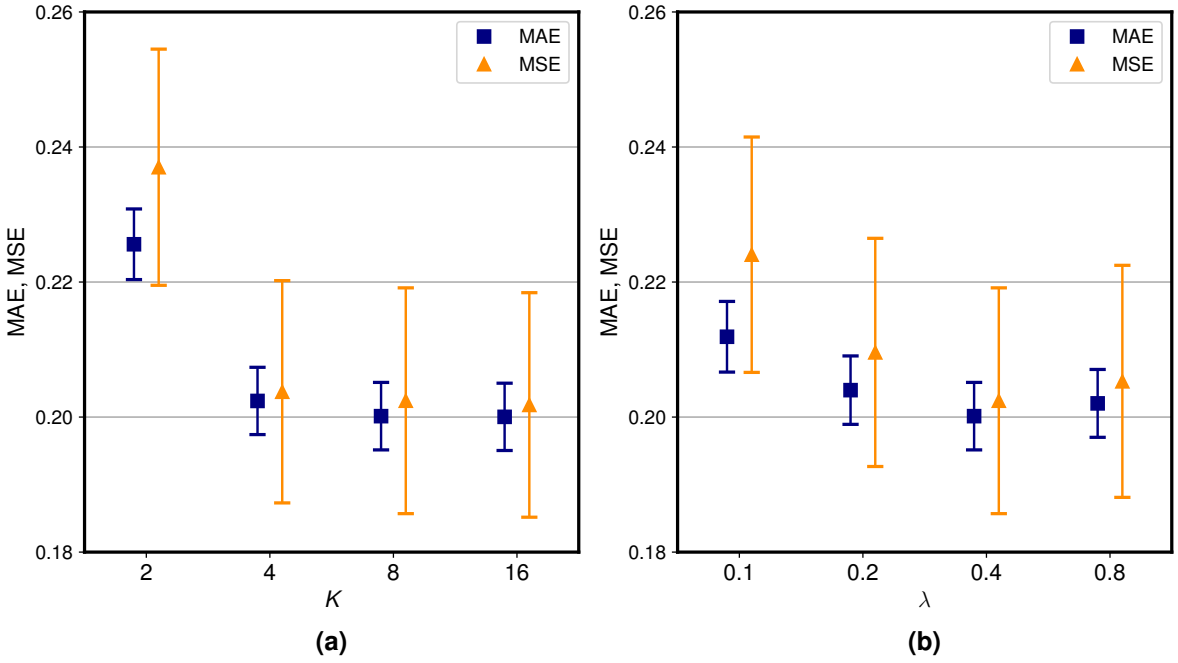


Figure F.3: Comparison of the predicted $\ln \gamma_i$ with UNIFAC 2.0 for different values of the latent dimension K (panel a) and the scale parameter of the Cauchy likelihood λ (panel b) evaluated on the complete horizon (27,287 data points for 2,603 binary mixtures) of the extrapolation study with unstudied components, cf. Fig. 41.

F.4 Extrapolation of Unseen Pair-Interaction Parameters

Fig. F.4 shows the selected group combinations for the extrapolation study. All data points requiring the respective a_{mn} were omitted from the training and used as a test set for each group combination. Since the considered a_{mn} are needed with varying frequencies to predict the binary mixtures of the experimental database, the resulting test sets fluctuate in the number of data points and mixtures. Table F.2 gives a detailed overview of all 100 test sets.

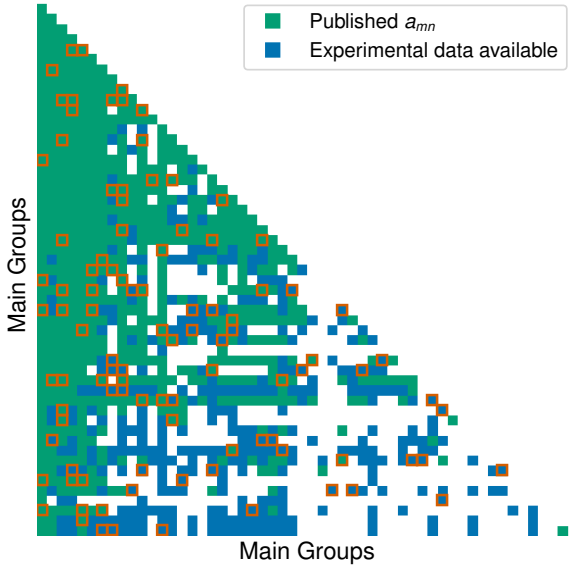


Figure F.4: Matrix of existing UNIFAC parameters (a_{mn}) of the public UNIFAC 1.0 model [12] (green) alongside additional group combinations for which experimental data are available (blue). Group-combinations that have been selected for the extrapolation study are highlighted by orange frames, cf. Table F.2.

In this extrapolation study, interaction parameters of UNIFAC 1.0 are available for 62 out of the 100 selected group combinations. However, the availability of these parameters does not guarantee the predictability of all binary mixtures within the test set, as they may need other necessary interaction parameters. To address this distinction, Table F.2 categorizes the data into two groups: those predictable with both UNIFAC 1.0 and UNIFAC 2.0 ("UNIFAC 1.0 horizon") and those exclusive to UNIFAC 2.0 ("UNIFAC 2.0 only"). Consequently, the prediction of the remaining 38 test sets can solely be carried out with UNIFAC 2.0.

Table F.2: Test sets evaluated for predicting interaction parameters a_{mn}. Each set is categorized into two groups: "UNIFAC 1.0 horizon" and "UNIFAC 2.0 only". The structural group identifiers $(m-n)$ are identical to UNIFAC 1.0 [12]. The table lists the number of data points (N_{data}) and binary mixtures (N_{mix}) for each set. It also includes the mixture-wise mean absolute errors, $\text{MAE}_{\text{mix}}^{1.0}$ and $\text{MAE}_{\text{mix}}^{2.0}$, for both UNIFAC methods.

$m-n$	UNIFAC 1.0 horizon				UNIFAC 2.0 only		
	N_{data}	N_{mix}	$\text{MAE}_{\text{mix}}^{2.0}$	$\text{MAE}_{\text{mix}}^{1.0}$	N_{data}	N_{mix}	$\text{MAE}_{\text{mix}}^{2.0}$
1-17	1017	134	0.32	0.22	92	71	0.18
1-29	392	52	0.32	0.18	73	25	0.32
1-32	685	143	0.40	0.11	89	33	0.18
1-49	482	32	0.36	0.16	43	6	0.22
1-55	1034	135	1.31	0.18	202	55	0.74
2-8	196	39	0.24	0.33	11	7	0.28
2-39	259	49	0.16	0.18			
2-45					151	10	0.11
3-11	2064	308	0.14	0.16	110	29	0.19
3-15	442	43	0.26	0.27	73	7	0.27
3-25	4305	535	0.23	0.34	292	108	0.19
3-30	689	22	0.17	0.11	32	1	0.05
3-32	135	51	0.07	0.09	47	24	0.18
3-39	448	17	0.17	0.20			
3-42	65	3	0.15	1.36	89	4	0.19
3-43	46	1	0.13	0.15	86	2	0.12
4-6	415	19	0.21	0.15	12	1	0.02
4-11	1221	132	0.11	0.13	76	13	0.18
4-12	102	7	0.17	0.12			
4-48					75	11	0.21
4-49	95	6	0.05	0.07			
5-6	1473	39	0.28	0.18	26	4	0.24
5-34	112	33	0.35	0.42	90	20	0.26
5-49	149	5	0.20	0.10			
5-55	35	11	0.16	0.20			

Table F.2 continued.

$m-n$	UNIFAC 1.0 horizon				UNIFAC 2.0 only		
	N_{data}	N_{mix}	$\text{MAE}^{2.0}_{\text{mix}}$	$\text{MAE}^{1.0}_{\text{mix}}$	N_{data}	N_{mix}	$\text{MAE}^{2.0}_{\text{mix}}$
5-84	711	100	0.19	0.13	280	61	0.18
6-28	37	1	0.36	0.23			
6-30	36	1	0.19	0.14			
6-32	53	2	0.29	0.05			
7-27	94	15	0.34	0.71	2	2	0.09
7-39	341	3	0.06	0.10			
7-55	15	1	0.11	0.36			
7-85					194	19	0.92
8-11	99	9	0.12	0.14			
8-20	51	6	1.41	2.57			
8-28	1	1	0.02	0.03			
8-37					1	1	0.13
8-38					30	15	0.17
8-40					26	18	0.25
8-85					6	1	2.77
9-10	300	33	0.10	0.10			
9-11	995	126	0.12	0.13	6	2	0.04
9-20	852	42	0.19	0.18	9	1	0.23
9-21	520	54	0.30	0.30	25	2	0.11
9-24	251	14	0.10	0.13			
9-29	2	2	0.10	0.08			
9-38					38	20	0.10
9-39	114	10	0.09	0.14			
9-40					131	24	0.24
10-30					6	1	1.23
10-50					28	15	0.10
11-12	54	8	0.08	0.08	3	1	0.06
11-15	377	1	0.09	0.03			

Table F.2 continued.

$m-n$	UNIFAC 1.0 horizon				UNIFAC 2.0 only		
	N_{data}	N_{mix}	$\text{MAE}^{2.0}_{\text{mix}}$	$\text{MAE}^{1.0}_{\text{mix}}$	N_{data}	N_{mix}	$\text{MAE}^{2.0}_{\text{mix}}$
11-30	80	5	0.28	0.25			
11-41	394	123	0.20	0.22	5	3	0.33
11-48					45	28	0.39
12-19	12	1	0.09	0.03			
13-26	22	12	0.15	0.13	27	9	0.11
13-34	10	9	0.08	0.07	96	21	0.12
13-41	587	116	0.19	0.21	33	11	0.59
13-85					1594	315	0.15
14-19	23	5	0.36	0.26			
14-35	15	1	0.26	0.28			
14-41					24	22	0.29
14-43	27	3	0.14	0.15			
15-24	52	2	0.09	0.10			
15-49					9	1	0.07
16-32					1	1	0.17
16-34					10	10	0.10
18-25	9	9	0.07	0.06			
18-32					3	3	0.12
18-38	32	3	0.22	0.24			
18-48					31	1	0.42
19-21	150	21	0.16	0.20	1	1	0.24
19-35					40	1	0.27
20-33	103	2	0.14	0.13			
20-34					12	4	0.10
20-46	154	3	0.39	0.37			
22-55					5	1	0.18
23-25	25	3	0.06	0.07			
23-30	14	1	0.26	0.17			

Table F.2 continued.

$m-n$	UNIFAC 1.0 horizon				UNIFAC 2.0 only		
	N_{data}	N_{mix}	$\text{MAE}_{\text{mix}}^{2.0}$	$\text{MAE}_{\text{mix}}^{1.0}$	N_{data}	N_{mix}	$\text{MAE}_{\text{mix}}^{2.0}$
23-45					27	4	0.09
24-45					2	1	0.04
25-39	35	2	0.45	0.45			
25-46					2	2	0.21
26-30					2	1	0.22
27-38					193	104	0.11
28-37	35	1	0.09	0.04			
30-50					2	1	0.05
31-32					1	1	0.44
31-47	36	1	0.10	0.21	43	3	0.16
32-50					2	2	0.64
33-38					43	8	0.09
35-37					3	3	0.59
38-47					23	15	0.13
39-47					9	1	0.02
40-41					160	70	0.23
41-42					3	2	0.22
41-51					9	2	0.28
47-48					20	7	0.18

F.5 Symmetric UNIFAC Model

In the following, a modification to the UNIFAC method by considering the symmetry of pair-interaction energies, denoted as $U_{mn} = U_{nm}$, is described. This contrasts with the approaches of UNIFAC 1.0 and 2.0, which directly optimize asymmetric pair-interaction parameters ($a_{mn} \neq a_{nm}$) that are derived as follows:

$$a_{mn} = U_{mn} - U_{nn} \tag{F.1}$$

$$a_{nm} = U_{nm} - U_{mm} \tag{F.2}$$

This variant is called *UNIFAC 2.0 sym* and optimizes the symmetric interaction energies, aligning with the physical consistency highlighted in Chapter 5.1. The predictive performance of this approach is depicted in Figure F.5, comparing the mean absolute error (MAE) and mean squared error (MSE) for both UNIFAC 2.0 and UNIFAC 2.0 sym across the extensive data set of 18,715 binary mixtures.

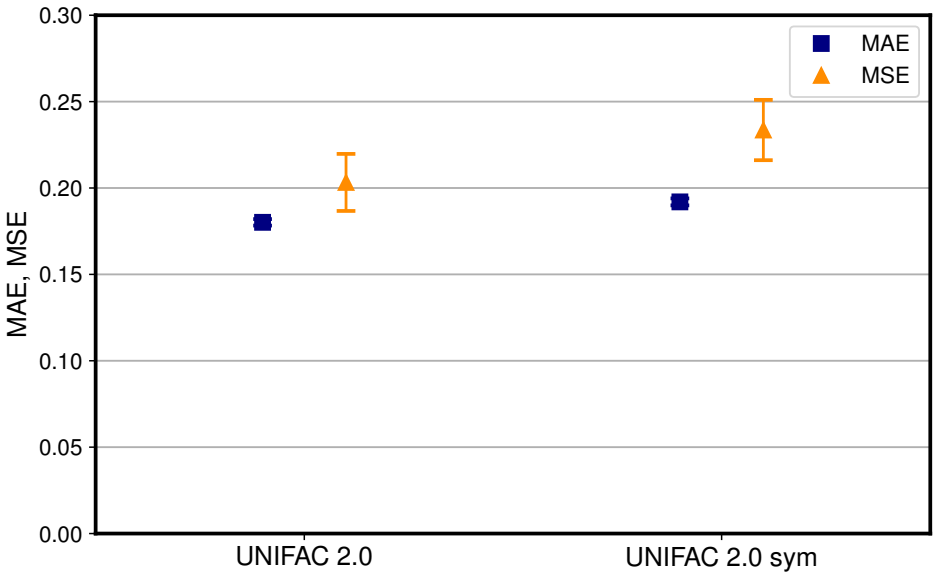


Figure F.5: Mean absolute error (MAE) and mean squared error (MSE) of the predicted $\ln \gamma_i$ with UNIFAC 2.0 and a model variant enforcing symmetric interaction energies (UNIFAC 2.0 sym). The whole binary data set was considered, comprising 224,562 data points for 18,715 binary mixtures. Error bars denote standard errors of the means.

Although the symmetric model offers greater physical consistency, its reduced flexibility slightly impacts prediction accuracy. Therefore, the primary focus is on UNIFAC 2.0.

G Supporting Information for Chapter 5.2.2

G.1 Modified UNIFAC Parameterization

Fig. G.1a provides a visualization of the pair-interaction parameters included in the standard mod. UNIFAC 1.0 model [13], as well as those that can be additionally fitted using the experimental database discussed in the "Data" section. Additionally, Fig. G.1b illustrates the distribution of the experimental data points considered here associated with each main group combination, highlighting the extent to which the respective pair-interaction parameters are supported by the available data.

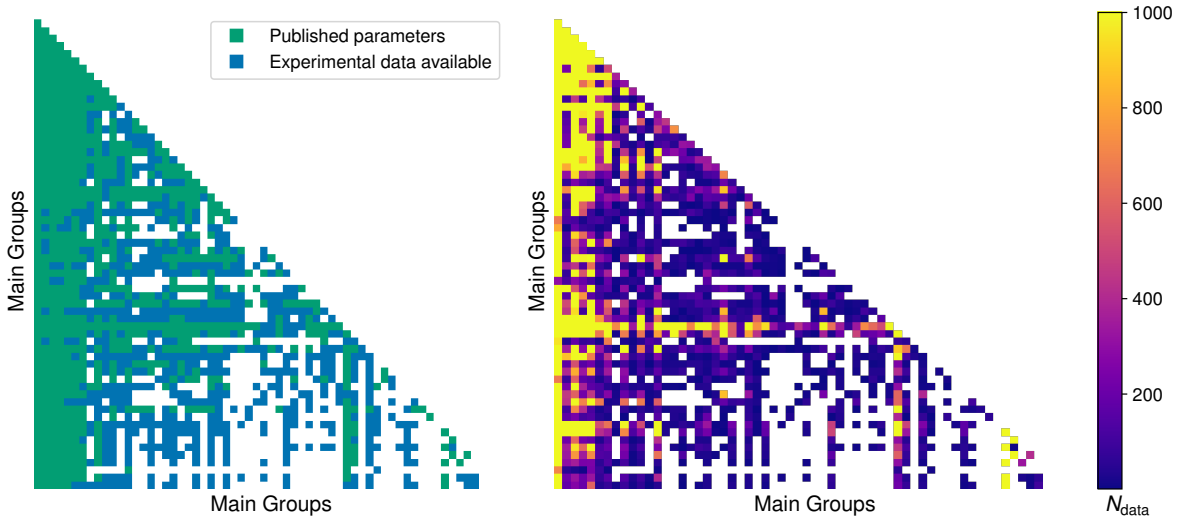


Figure G.1: (a) Representation of the published mod. UNIFAC 1.0 pair-interaction parameters [13] (green) and the ones that could additionally be fitted using the experimental data from the DDB [142] (blue). (b) Heatmap of the number of experimental data points ($\ln \gamma_i$ and h^{E}) from the DDB requiring specific main group combinations.

The heatmaps reveal a strong disparity in data coverage. For example, while 216 group combinations (11% of the matrix) are associated with more than 1,000 data points, 248

group combinations (13%) are only relevant in 20 or fewer data points. Even worse, 594 parameters (30%) do not appear in the available experimental data and can not be directly fitted. This pronounced heterogeneity underscores the challenges of parameter fitting and emphasizes the importance of models like mod. UNIFAC 2.0 that efficiently use the available experimental data to fill the gaps.

Fig. G.2 extends Fig.G.1a by incorporating the interaction parameters available in the commercial UNIFAC-Consortium model [141]. It is important to note that the UNIFAC-Consortium version includes more main groups than the 63 considered in the public version, which have been omitted for consistency here.

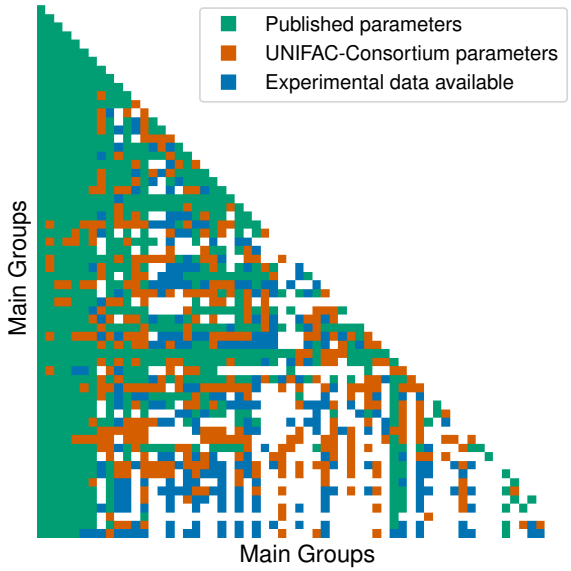


Figure G.2: Matrix of existing pair-interaction parameters of the public mod. UNIFAC 1.0 model [13] (green), supplemented by those of the commercial UNIFAC-Consortium version [141] (orange). Furthermore, group combinations are marked for which data are available, but no parameters have yet been fitted (blue).

While the UNIFAC-Consortium model provides a substantially broader scope than the public mod. UNIFAC 1.0, the heatmap in Fig. G.2 still highlights significant gaps in the interaction parameter matrix. These gaps are primarily due to the limited availability of experimental training data, emphasizing the critical need for extrapolative methods such as mod. UNIFAC 2.0. Unlike traditional models, mod. UNIFAC 2.0 can predict these missing parameters, thus bridging the gaps in the interaction parameter space. However, since the parameter tables for the UNIFAC-Consortium model are proprietary, a direct evaluation or comparison of its predictive accuracy could not be performed.

G.2 Extrapolation to Multi-Component Mixtures

Despite the absence of multi-component mixture data during the training of mod. UNIFAC 2.0, the physical principles of its framework allow it to make reliable predictions for such systems. To illustrate this capability, Fig. G.3 shows isothermal vapor-liquid phase diagrams for two ternary mixtures from the "mod. UNIFAC 2.0 only" data set, i.e., mixtures for which mod. UNIFAC 1.0 cannot be applied due to missing pair-interaction parameters.

In these examples, the temperature and liquid-phase composition (shown as blue symbols in Fig. G.3) were used as inputs to mod. UNIFAC 2.0. The model predicted the activity coefficients from which the corresponding vapor-phase composition at equilibrium (filled orange symbols) could be calculated using the extended Raoult's law, cf. Eq. (43). For comparison, the experimental vapor-phase compositions are also shown (open orange symbols). The predictions are in excellent agreement with the experimental data, demonstrating the model's suitability for describing multi-component mixtures.

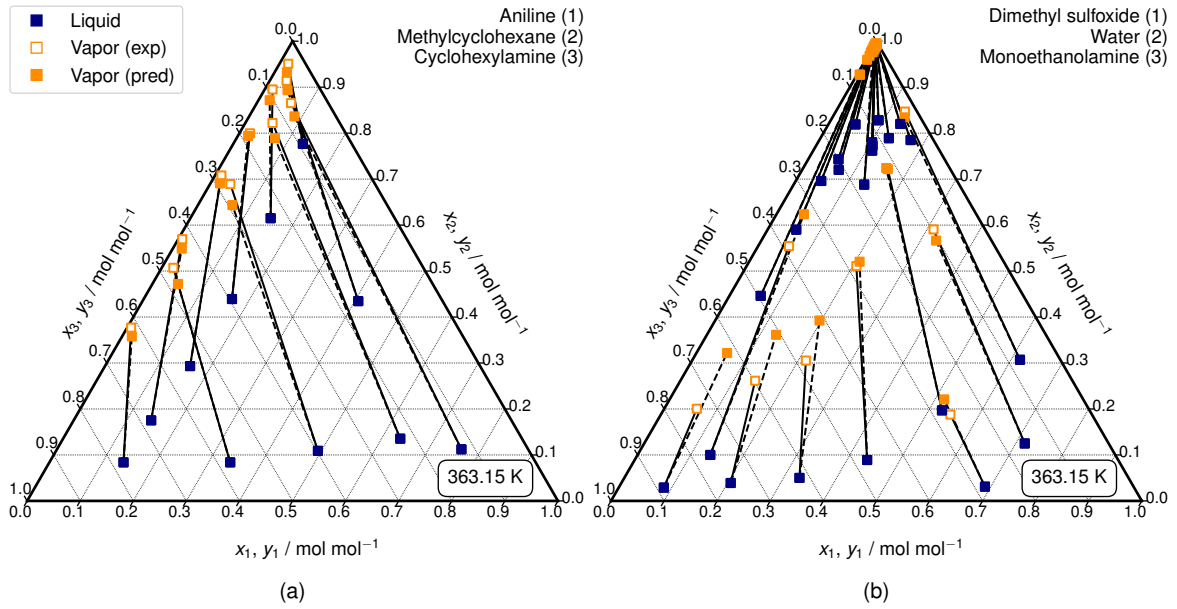


Figure G.3: Prediction of isothermal vapor-liquid phase diagrams for ternary mixtures with mod. UNIFAC 2.0 (pred) and comparison to experimental data (exp) from the DDB. The temperature and the composition of the liquid phase were specified, and the composition of the corresponding vapor phase in equilibrium was predicted. Solid lines are experimental conodes, dashed lines are predicted conodes. Mod. UNIFAC 1.0 is not applicable to the mixtures shown.

G.3 Prediction of Excess Enthalpies for Unseen Components

The extrapolation capability of mod. UNIFAC 2.0 for mixtures containing unseen components was evaluated by randomly selecting 100 components and training the model on all available data ($\ln \gamma_i$ and h^{E}) for those mixtures where these components do not occur. The retained mixtures served as the test set. Fig. G.4 presents results for predicting h^{E} on this test set in a box plot, analogous to Fig. 49, which focuses on $\ln \gamma_i$.

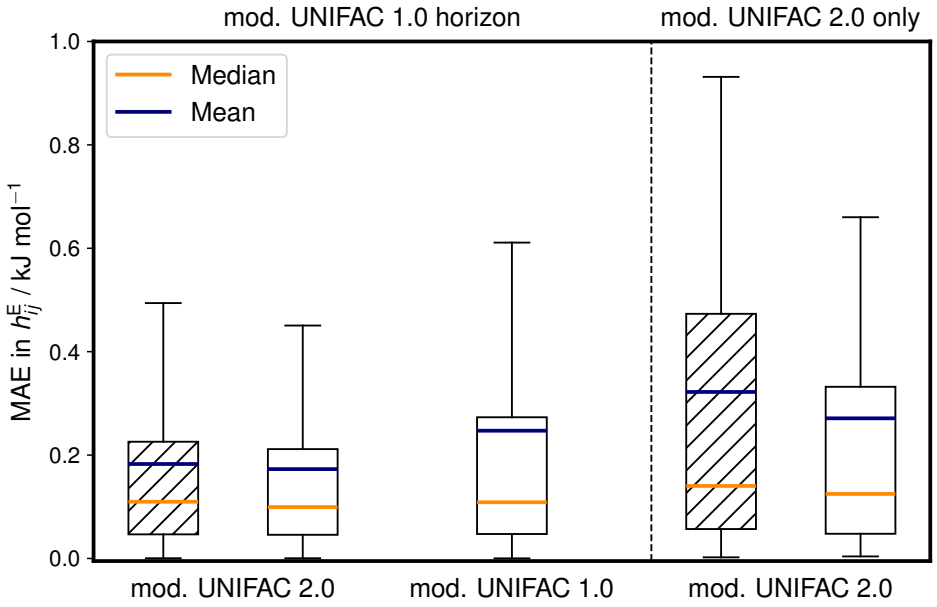


Figure G.4: Mean absolute error (MAE) of the predicted h^{E} of mixtures containing unseen components with mod. UNIFAC 2.0 (shaded boxes). For comparison, the results of mod. UNIFAC 2.0 trained on all experimental data and mod. UNIFAC 1.0 are also shown (plain boxes). The "mod. UNIFAC 1.0 horizon" comprises 11,906 data points for 473 binary mixtures, while an additional 1,289 experimental data points for 71 binary mixtures could only be predicted with mod. UNIFAC 2.0 ("mod. UNIFAC 2.0 only"). The boxes represent the interquartile ranges (IQR), and the whiskers extend to the last data points within 1.5 times the IQR from the box edges.

On the "mod. UNIFAC 1.0 horizon", mod. UNIFAC 2.0 not only outperforms mod. UNIFAC 1.0 but also achieves comparable prediction accuracies whether trained on all experimental data or tasked with true extrapolation, demonstrating its robust extrapolation capabilities. On the "mod. UNIFAC 2.0 only" set, slightly higher MAE values are observed. However, the method still provides reasonable predictions for most mixtures, as evidenced by the low median, underscoring its applicability in scenarios where mod. UNIFAC 1.0 cannot be applied.

G.4 Extrapolation to Unseen Pair-Interaction Parameters

Fig. G.5 shows the 100 main group combinations randomly selected for the second extrapolation study, cf. Fig. 50. For each combination, all $\ln \gamma_i$ and h^{E} data associated with the corresponding interaction parameters were removed from the training set and used exclusively for testing. The frequency of these group combinations varies in the experimental database, resulting in test sets of different sizes and compositions. A comprehensive summary of all 100 test sets, including the number of data points and mixtures, is provided in Tables G.1 (for $\ln \gamma_i$) and G.2 (for h^{E}). These tables also include MAEs for each individual test set.

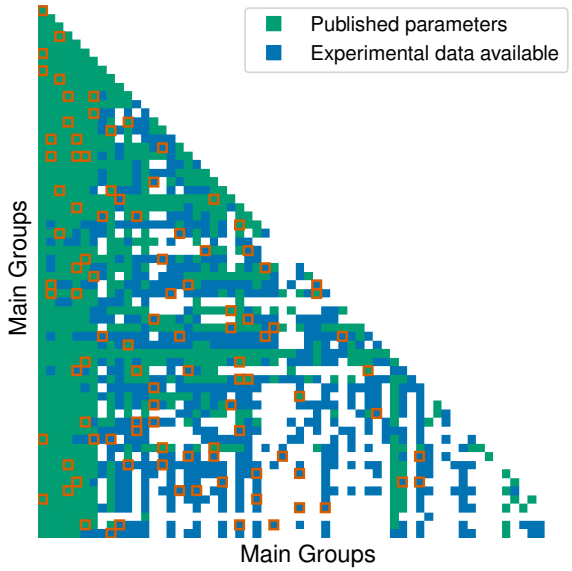


Figure G.5: Matrix of available pair-interaction parameters of the mod. UNIFAC 1.0 model [13] (green) alongside additional group combinations for which experimental data are available [142] (blue). Group combinations that have been selected for the extrapolation study in this chapter are highlighted by orange frames, cf. Tables G.1 and G.2.

In this extrapolation study, 53 of the selected group combinations were not parameterized in mod. UNIFAC 1.0. Moreover, even when these parameters are available, they do not guarantee that all binary mixtures within the test sets can be predicted, as additional interaction parameters might also be required. To address this distinction, Tables G.1 and G.2 categorize the data into two groups: mixtures that can be predicted by both mod. UNIFAC 1.0 and mod. UNIFAC 2.0 (referred to as the "mod. UNIFAC 1.0 horizon") and those that can only be predicted by mod. UNIFAC 2.0 ("UNIFAC 2.0 only"). Since certain group combinations have data for either $\ln \gamma_i$ or h^{E} (but not both), they appear in only one of the following tables.

Table G.1: Test sets for $\ln \gamma_i$ evaluated for predicting interaction parameters. Each set is categorized into two groups: "mod. UNIFAC 1.0 horizon" and "mod. UNIFAC 2.0 only". The main group identifiers ($m-n$) are identical to mod. UNIFAC 1.0 [13]. The table lists the number of data points (N_{data}) and binary mixtures (N_{mix}) for each set. It also includes the mixture-wise mean absolute errors, $\text{MAE}_{\text{mix}}^{1.0}$ and $\text{MAE}_{\text{mix}}^{2.0}$, for both mod. UNIFAC methods.

$m-n$	mod. UNIFAC 1.0 horizon				mod. UNIFAC 2.0 only		
	N_{data}	N_{mix}	$\text{MAE}_{\text{mix}}^{2.0}$	$\text{MAE}_{\text{mix}}^{1.0}$	N_{data}	N_{mix}	$\text{MAE}_{\text{mix}}^{2.0}$
1-2	21164	3021	0.18	0.22	1617	535	0.21
1-7	19331	680	1.18	1.35	514	58	1.27
1-9	19557	1557	0.17	0.16	976	153	0.38
1-55	672	26	0.06	0.08	1447	58	0.18
1-90	1860	268	0.24	0.47	1115	224	0.26
2-17	92	20	0.09	0.91	15	13	0.25
2-19	741	275	0.19	0.14	131	101	0.11
2-34	206	21	0.08	0.05	85	25	0.25
2-35	37	15	0.16	0.16	19	10	0.41
3-5	10550	795	0.23	0.51	367	63	0.22
3-10	672	69	0.19	0.2	192	71	0.28
3-23	45	4	0.24	0.02	26	9	0.14
3-28	82	7	0.05	0.05	3	3	0.27
4-12	102	7	0.13	0.1			
4-15	136	15	0.1	0.13	70	4	0.19
4-84	1971	284	0.2	0.18	632	145	0.18
4-89	450	70	0.37	0.34	109	37	0.18
5-17	97	6	0.32	0.13			
5-19	994	70	0.17	0.17	29	8	0.14
5-25	818	29	0.11	0.11	25	5	0.08
5-32	50	14	0.06	0.06			
5-35	200	8	0.29	0.21	48	5	0.28
5-87	143	35	0.05	0.07	42	7	0.08
6-19	269	8	0.17	0.12	7	3	0.11

Table G.1 continued.

$m-n$	mod. UNIFAC 1.0 horizon				mod. UNIFAC 2.0 only		
	N_{data}	N_{mix}	$\text{MAE}_{\text{mix}}^{2.0}$	$\text{MAE}_{\text{mix}}^{1.0}$	N_{data}	N_{mix}	$\text{MAE}_{\text{mix}}^{2.0}$
6-31	69	1	0.22	0.17			
6-43	372	7	0.13	0.14			
6-52	88	1	0.09	0.03			
6-98	56	10	0.35	0.46	13	3	0.31
7-12	136	14	0.56	0.31			
7-14	1313	34	0.85	0.46			
7-33	166	27	1.06	0.98	6	1	0.62
7-55					21	2	0.32
8-26					4	4	0.42
8-40					26	18	0.31
9-16	102	28	0.09	0.19	11	5	0.11
9-23	48	2	0.06	0.06			
9-55					260	8	0.1
9-99					83	13	0.17
10-24					11	1	0.41
10-98					30	6	0.2
11-15	377	1	0.06	0.03			
11-41	346	105	0.21	0.27	53	21	0.24
11-84					517	95	0.38
12-44	135	5	0.26	0.23			
12-53					11	1	0.38
14-22					11	3	0.17
14-52					6	1	0.39
15-18					14	1	0.12
15-31					75	2	0.26
16-35					1	1	0.37
17-28					1	1	0.19
17-40					42	26	0.35

Table G.1 continued.

$m-n$	mod. UNIFAC 1.0 horizon				mod. UNIFAC 2.0 only		
	N_{data}	N_{mix}	$\text{MAE}^{2.0}_{\text{mix}}$	$\text{MAE}^{1.0}_{\text{mix}}$	N_{data}	N_{mix}	$\text{MAE}^{2.0}_{\text{mix}}$
18-34					4	4	0.27
18-61					31	1	0.26
19-89					72	13	0.19
20-30					40	3	0.68
21-24	228	11	0.14	0.1	22	1	0.14
21-56	9	2	0.4	0.03			
21-61					3	3	0.29
22-87					24	7	0.15
23-37	36	3	0.09	0.05	4	2	1.8
23-48					13	11	0.32
24-27	18	1	0.6	0.49			
24-43	192	2	0.02	0.03			
24-45	127	1	0.14	0.01			
24-53					42	2	0.05
24-98					30	6	0.3
25-30					32	1	0.05
25-38					40	8	0.1
25-45	19	1	0.12	0.08	3	1	0.76
26-85					184	44	0.18
26-90					34	7	0.11
27-32					3	2	0.27
27-40					182	101	0.21
28-39					1	1	0.25
28-98					15	3	0.43
29-61					24	2	0.08
31-47	10	1	0.08	0.02			
33-34					1	1	0.15
33-35					1	1	0

Table G.1 continued.

$m-n$	mod. UNIFAC 1.0 horizon				mod. UNIFAC 2.0 only		
	N_{data}	N_{mix}	$\text{MAE}^{2.0}_{\text{mix}}$	$\text{MAE}^{1.0}_{\text{mix}}$	N_{data}	N_{mix}	$\text{MAE}^{2.0}_{\text{mix}}$
34-91					78	13	0.33
36-40					3	1	0.19
39-44	128	2	0.82	0.84			
43-87					50	11	0.11
45-61					17	2	0.4
45-87					24	7	0.11

Table G.2: Test sets for h^{E} evaluated for predicting interaction parameters. Each set is categorized into two groups: "mod. UNIFAC 1.0 horizon" and "mod. UNIFAC 2.0 only". The main group identifiers ($m-n$) are identical to mod. UNIFAC 1.0 [13]. The table lists the number of data points (N_{data}) and binary mixtures (N_{mix}) for each set. It also includes the mixture-wise mean absolute errors, $\text{MAE}^{1.0}_{\text{mix}}$ and $\text{MAE}^{2.0}_{\text{mix}}$, for both mod. UNIFAC methods.

$m-n$	mod. UNIFAC 1.0 horizon				mod. UNIFAC 2.0 only		
	N_{data}	N_{mix}	$\text{MAE}^{2.0}_{\text{mix}}$	$\text{MAE}^{1.0}_{\text{mix}}$	N_{data}	N_{mix}	$\text{MAE}^{2.0}_{\text{mix}}$
1-2	11734	599	0.15	0.26	392	28	0.31
1-7	15366	190	0.55	0.54	618	12	0.36
1-9	19871	611	0.2	0.19	797	40	0.27
1-55	1664	35	0.15	0.13	3442	114	0.22
1-90	349	19	0.86	0.46	3	1	0.03
2-17	114	8	0.16	0.27			
2-19	281	17	0.2	0.22	11	2	0.76
2-34	51	6	0.05	0.12	15	1	0.49
2-35	23	3	0.26	0.34			
3-5	10942	293	0.36	0.78	599	12	0.55
3-10	631	39	0.27	1.18	45	2	0.12
3-23	168	10	0.28	0.13	4	2	1.41
3-28	31	3	0.09	0.07	2	1	0.18

Table G.2 continued.

$m-n$	mod. UNIFAC 1.0 horizon				mod. UNIFAC 2.0 only		
	N_{data}	N_{mix}	$\text{MAE}^{2.0}_{\text{mix}}$	$\text{MAE}^{1.0}_{\text{mix}}$	N_{data}	N_{mix}	$\text{MAE}^{2.0}_{\text{mix}}$
4-12	34	2	0.01	0.01			
4-15	116	7	0.09	0.49	18	1	1.65
4-84	64	6	0.11	0.1	54	3	0.18
4-89	8	1	0.03	0.04	54	3	0.18
5-17	309	8	0.37	0.18	18	2	2.25
5-19	1118	34	0.19	0.25	21	1	0.74
5-25	513	24	0.23	0.4			
5-32	37	2	0.32	0.38			
5-35	633	18	0.33	0.24	149	7	0.74
5-44					83	9	0.16
6-19	135	5	0.07	0.15			
6-31	160	1	0.21	0.02			
6-43	286	3	0.14	0.08	10	1	0.02
6-52	38	1	0.3	1.12			
7-12	53	3	0.29	0.43			
7-14	876	14	0.91	0.38	85	2	0.8
7-33	20	1	0.63	0.43			
9-16	364	20	0.32	0.33			
9-23	70	6	0.24	0.19			
9-55					378	16	0.06
10-24					76	5	0.11
11-15	204	9	0.21	0.14			
11-41	945	35	0.24	0.13	66	7	0.15
12-26					57	2	0.04
12-52					30	1	0.01
14-38					20	1	0.01
14-46					33	1	0.34
14-48					36	2	0.42

Table G.2 continued.

$m-n$	mod. UNIFAC 1.0 horizon				mod. UNIFAC 2.0 only		
	N_{data}	N_{mix}	$\text{MAE}_{\text{mix}}^{2.0}$	$\text{MAE}_{\text{mix}}^{1.0}$	N_{data}	N_{mix}	$\text{MAE}_{\text{mix}}^{2.0}$
15-18					33	4	0.1
15-31					8	1	1.84
15-56	40	1	0.23	0.12			
15-84					18	1	0.47
16-35					24	1	1.13
17-89					36	2	1.22
18-34					14	2	0.64
18-44					33	1	3.86
21-24	556	18	0.14	0.09	68	2	0.39
23-39					15	1	0.04
23-48					3	1	1.04
24-43	603	9	0.16	0.16			
24-45	106	1	0.33	0.01			
24-53					281	5	0.09
25-38					84	6	0.07
25-45	8	4	0.17	0.27			
28-39					1	1	0.1
29-61					10	1	0.17
31-47	10	1	0.22	0.01			
31-85					21	1	0.19
31-91					9	1	0.19
33-35	126	5	0.72	0.45			
40-49					16	1	0.82
45-61					51	3	0.91

Fig. G.6 shows the average error scores for predicting h^{E} across all 100 test sets, analogous to Fig. 50, which focuses on $\ln \gamma_i$. As before, the performance of mod. UNIFAC 2.0

is compared to that of mod. UNIFAC 1.0 and the version of mod. UNIFAC 2.0 trained on the entire experimental database.

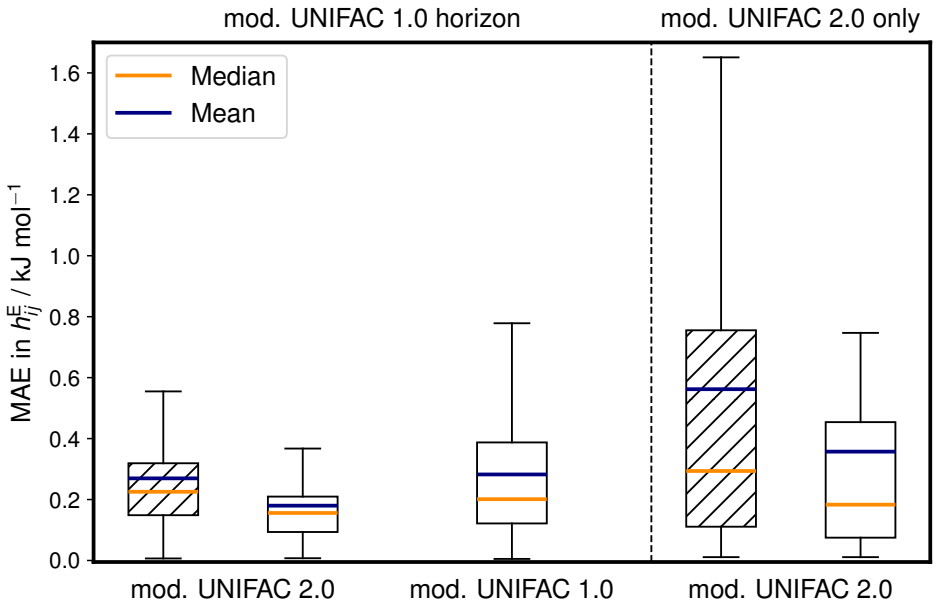


Figure G.6: Mean absolute error (MAE) of the predicted h^{E} with mod. UNIFAC 2.0 for 100 test sets, where all data points for which a specific main group combination is relevant were withheld during training (shaded boxes); cf. Table G.2 for numerical results. The results of mod. UNIFAC 2.0 trained on all experimental data and mod. UNIFAC 1.0 are shown for comparison (plain boxes). The boxes represent the interquartile ranges (IQR), and the whiskers extend to the last data points within 1.5 times the IQR from the box edges.

Comparing the predictions of mod. UNIFAC 2.0 with those of mod. UNIFAC 1.0 on the "UNIFAC 1.0 horizon" shows that the fitted pair-interaction parameters of mod. UNIFAC 2.0 outperform those of mod. UNIFAC 1.0, while its true predictions achieve comparable accuracy. When evaluating the true predictions of mod. UNIFAC 2.0 against the model trained on the entire experimental data set, a slight but expected decrease in accuracy is observed. Nevertheless, the differences remain moderate, underscoring the robustness of mod. UNIFAC 2.0 in extrapolating to unseen interaction parameters, a capability inherently lacking in mod. UNIFAC 1.0.

Declaration

This dissertation contains material that has been published previously or that is included in submitted publications. In the following, these publications are listed together with a statement on the contributions of the author of the present dissertation.

- N. Hayer, F. Jirasek, H. Hasse: Prediction of Henry's law constants by matrix completion, *AIChE Journal*, 68 (2022) e17753, DOI: 10.1002/ aic.17753.
  The author developed and trained the machine learning models. The author wrote the manuscript.

- O. Großmann, D. Bellaire, N. Hayer, F. Jirasek, H. Hasse: Database for liquid phase diffusion coefficients at infinite dilution at 298 K and matrix completion methods for their prediction, *Digital Discovery*, 1 (2022) 886-897, DOI: 10.1039/ D2DD00073C.
  The author developed the machine learning models and trained them on the database that was consolidated by the co-authors. He contributed to writing the manuscript.

- F. Jirasek, N. Hayer, R. Abbas, B. Schmid, H. Hasse: Prediction of parameters of group contribution models of mixtures by matrix completion, *Physical Chemistry Chemical Physics*, 25 (2022) 1054-1062, DOI: 10.1039/ D2CP04478A.
  The author contributed to developing and training the machine learning model. He contributed to writing the manuscript.

- N. Hayer, H. Hasse, F. Jirasek: Prediction of Temperature-Dependent Henry's Law Constants by Matrix Completion, *The Journal of Physical Chemistry B*, 129 (2024) 409-416, DOI: 10.1021/acs.jpcb.4c07196.
  The author developed and trained the machine learning model. The author wrote the manuscript.

- N. Hayer, T. Specht, J. Arweiler, D. Gond, H. Hasse, F. Jirasek: Prediction of Activity Coefficients by Similarity-Based Imputation using Quantum-Chemical Descriptors, *Physical Chemistry Chemical Physics*, 27 (2025) 4307-4315, DOI: 10.1039/D4CP04341C.
  The author developed the similarity-based approach together with Thomas Specht. The author wrote the manuscript.

- N. Hayer, T. Specht, J. Arweiler, H. Hasse, F. Jirasek: Similarity-Informed Matrix Completion Method for Predicting Activity Coefficients, *The Journal of Physical Chemistry A*, 129 (2025) 3141–3147, DOI: 10.1021/acs.jpca.4c08360 .
  The author developed the machine learning model together with Thomas Specht. The author wrote the manuscript.

- N. Hayer, T. Wendel, S. Mandt, H. Hasse, F. Jirasek: Advancing Thermodynamic Group-Contribution Methods by Machine Learning: UNIFAC 2.0, *Chemical Engineering Journal*, 504 (2024) 158667, DOI: 10.1016/j.cej.2024.158667 .
  The author developed and trained the machine learning model. The author wrote the manuscript.

- N. Hayer, H. Hasse, F. Jirasek: Modified UNIFAC 2.0 – A Group Contribution Method Completed with Machine Learning, *Industrial & Engineering Chemistry Research*, (2025), DOI: 10.1021/acs.iecr.5c00077 .
  The author developed and trained the machine learning model. The author wrote the manuscript.

Student Theses

The following student theses were prepared under the supervision of the author of the present doctoral thesis in the frame of his research:

- A. Karabulut: Prediction of Henry's Law Constants with Matrix Completion Methods. Bachelor thesis, Laboratory of Engineering Thermodynamics (LTD), TU Kaiserslautern (2021).
- J. Stüber: Prediction of Physicochemical Properties with Hybrid Approaches of Matrix Completion Methods and COSMO-RS. Project thesis, Laboratory of Engineering Thermodynamics (LTD), TU Kaiserslautern (2022).
- J. Arweiler: Prediction of Activity Coefficients at Infinite Dilution Using Hybrid Matrix Completion Methods with Quantum-Chemical Descriptors. Project thesis, Laboratory of Engineering Thermodynamics (LTD), TU Kaiserslautern (2022).
- M. Hoffmann: Incorporating Molecular Simulations in Machine Learning Approaches for Henry's Law Constants. Diploma thesis, Laboratory of Engineering Thermodynamics (LTD), TU Kaiserslautern (2022).
- R. Kostanzer: Active Learning of Matrix Completion Methods for the Prediction of Fluid Properties of Mixtures. Bachelor thesis, Laboratory of Engineering Thermodynamics (LTD), RPTU Kaiserslautern (2023).
- I. Reyre: Prediction of the Parameters of Matrix Completion Methods based on Pure-Component Descriptors. Diploma thesis, Laboratory of Engineering Thermodynamics (LTD), RPTU Kaiserslautern (2023).
- M. August: Active Learning of Matrix Completion Methods for the Prediction of Second Virial Coefficients of Mixtures. Study project, Laboratory of Engineering Thermodynamics (LTD), RPTU Kaiserslautern (2023).
- L. Felsing: Prediction of NRTL Parameters by Hybrid Matrix Completion Methods. Bachelor thesis, Laboratory of Engineering Thermodynamics (LTD), RPTU Kaiserslautern (2023).

Curriculum Vitae

Name:	Nicolas Hayer
Place of birth:	Wittlich
Nationality:	German

Education

2000 – 2004	Elementary School Hasborn
2004 – 2013	Peter-Wust-Gymnasium Wittlich
	Degree: Allgemeine Hochschulreife
2013 – 2018	Technische Universität Kaiserslautern
	Program: Bio- und Chemieingenieurwissenschaften
	Degree: Bachelor of Science
2017 – 2019	Technische Universität Kaiserslautern
	Program: Bio- und Chemieingenieurwissenschaften
	Degree: Master of Science

Professional

2020 – 2025	Research Associate at Laboratory of Engineering Thermodynamics (LTD)
	RPTU Kaiserslautern, Prof. Dr.-Ing. Hans Hasse
2023	Visiting Researcher at Department of Computer Science
	University of California, Irvine.