Graph neural networks for temperature-dependent activity coefficient prediction of solutes in ionic liquids

Rittig, Jan G.; Ben Hicham, Karim; Schweidtmann, Artur M.; Dahmen, Manuel; Mitsos, Alexander

DOI
10.1016/j.compchemeng.2023.108153

Publication date
2023

Document Version
Final published version

Published in
Computers and Chemical Engineering

Computers and Chemical Engineering 171 (2023) 108153

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/cace

Graph neural networks for temperature-dependent activity coefficient prediction of solutes in ionic liquids

Jan G. Rittig [a], Karim Ben Hicham [a], Artur M. Schweidtmann [b], Manuel Dahmen [c], Alexander Mitsos [d,a,c,*]

[a] Process Systems Engineering (AVT.SVT), RWTH Aachen University, Forckenbeckstraße 51, 52074 Aachen, Germany
[b] Delft University of Technology, Department of Chemical Engineering, Van der Maasweg 9, Delft 2629 HZ, The Netherlands
[c] Forschungszentrum Jülich GmbH, Institute of Energy and Climate Research IEK-10 – Energy Systems Engineering, Wilhelm-Johnen-Str., 52425 Jülich, Germany
[d] JARA Center for Simulation and Data Science (CSD), Aachen, Germany

ARTICLE INFO

Dataset link: https://git.rwth-aachen.de/avt-svt/public/gnn_gamma_IL

Keywords:
Graph learning
Machine learning
Green solvents
Activity coefficient prediction
Ionic liquids

ABSTRACT

Ionic liquids (ILs) are important solvents for sustainable processes and predicting activity coefficients (ACs) of solutes in ILs is needed. Recently, matrix completion methods (MCMs), transformers, and graph neural networks (GNNs) have shown high accuracy in predicting ACs of binary mixtures, superior to well-established models, e.g., COSMO-RS and UNIFAC. GNNs are particularly promising here as they learn a molecular graph-to-property relationship without pretraining, typically required for transformers, and are, unlike MCMs, applicable to molecules not included in training. For ILs, however, GNN applications are currently missing. Herein, we present a GNN to predict temperature-dependent infinite dilution ACs of solutes in ILs. We train the GNN on a database including more than 40,000 AC values and compare it to a state-of-the-art MCM. The GNN and MCM achieve similar high prediction performance, with the GNN additionally enabling high-quality predictions for ACs of solutions that contain ILs and solutes not considered during training.

1. Introduction

Ionic liquids (ILs) have gained great interest in many chemical engineering applications due to their attractive physico-chemical properties such as negligible vapor pressure (Seddon, 1997; Rogers and Seddon, 2003; Berthod et al., 2018). A wide variety of ILs can be formed by combining anions, cations, and other structural groups, enabling tuning of ILs with respect to specific properties and resulting in a design space containing over one million potential binary ILs (Rogers and Seddon, 2003). Applications of ILs as solvents, catalysts, or electrolytes are vast, cf. Seddon (1997), Rogers and Seddon (2003), Plechkova and Seddon (2008), Lei et al. (2017) including separation processes (Brennecke and Maginn, 2001; Lei et al., 2014), biomass conversion (Zhang et al., 2017), and batteries (Galiński et al., 2006). With these IL applications typically involving mixtures, the activity coefficient (AC) is one of the most important properties as it accounts for intermolecular interactions and thus enables to model non-ideality of mixtures with ILs, cf. Lei et al. (2014), Han and Armstrong (2007), Zeng et al. (2017), Chen et al. (2021). For using ILs as solvents, particularly, the infinite dilution AC approximating non-ideal behavior of solutions with high solvent and low solute concentrations is highly relevant. To design ILs with application-specific properties, the AC therefore needs to be considered. For exploring the large design space of potential ILs, computer-aided molecular design methods can be utilized, cf. Karunanithi and Mehrkesh (2013), Peng et al. (2017), Song et al. (2018, 2019), Wang et al. (2018). In relation to the large design space, however, the availability of experimental AC data is very limited. Additionally, determining the AC experimentally for many IL candidates would be time-prohibitive and expensive. Rather, models that enable fast and high-quality AC predictions are desired as they enable to explore a large number of IL candidates and are thus essential building blocks for computer-aided IL design.

Model-based AC prediction with thermodynamic equation-of-state methods and approaches like UNIFAC (Fredenslund et al., 1975) and COSMO-RS (Klamt, 1995) is well established in chemical engineering. Such classical AC models have also been adapted for mixtures containing ILs. For example, UNIFAC which maps the structural groups of molecules to ACs has been extended to ILs (Chen et al., 2021; Song et al., 2018; Lei et al., 2009, 2012; Roughton et al., 2012; Dong et al., 2020), see, e.g., UNIFAC-IL (Song et al., 2016). COSMO-RS, rooted in statistical thermodynamics, has also been applied and extended to the prediction of ACs of solutes in ILs (Chen et al., 2021; Song et al., 2016; Han et al., 2018). For instance, Song et al. presented

* Corresponding author at: Process Systems Engineering (AVT.SVT), RWTH Aachen University, Forckenbeckstraße 51, 52074 Aachen, Germany.
E-mail address: amitsos@alum.mit.edu (A. Mitsos).

https://doi.org/10.1016/j.compchemeng.2023.108153
Received 25 June 2022; Received in revised form 7 December 2022; Accepted 23 January 2023
Available online 26 January 2023

a modified COSMO-RS model calibrated to different molecular solute families by considering large amounts of experimental data which has been shown to produce improved results for IL solutions compared to standard COSMO-RS (Song et al., 2016). However, classical AC prediction models also come with limitations. UNIFAC can only be applied to molecules composed solely of those structural groups that have already been parameterized based on experimental data, and COSMO-RS has been found to have limited prediction accuracy for ACs of solutes in ILs (Chen et al., 2021; Song et al., 2016).

Recently, methods from machine learning (ML), namely matrix completion methods (MCMs) (Jirasek et al., 2020; Damay et al., 2021), transformers (Winter et al., 2022), and graph neural networks (GNNs) (Sanchez Medina et al., 2022; Felton et al., 2022), have been actively investigated as alternative approaches for predicting ACs, with MCMs also applied to IL solutions (Chen et al., 2021). Generally, ML approaches aim to learn molecular structure-to-property relationships directly from data, cf. (Butler et al., 2018; Muratov et al., 2020; Rittig et al., 2022b).

As the name suggests, MCMs encode solvents and solutes in a matrix, with AC of a specific solvent–solute combination as entries. Since only few entries are filled by available experimental data, the matrix is typically sparse and has to be completed. Recently, MCMs based on collaborative filtering (He et al., 2017) have been utilized to AC prediction (Jirasek et al., 2020; Damay et al., 2021; Chen et al., 2021), relying on the concept that similarities between different rows and different columns, i.e., patterns, deduced solely from the given AC entries of a sparse solute-solvent matrix can be used to fill in the missing entries. For example, Jirasek et al. (2020) utilized a MCM for predicting infinite dilution ACs at ambient temperature which performed favorably in comparison to UNIFAC. Damay et al. (2021) extended this MCM model (Jirasek et al., 2020) to also include a temperature-dependency of the infinite dilution AC by utilizing the Gibbs-Helmholtz relation. For the prediction of temperature-dependent infinite dilution ACs of solutes in ILs, Chen et al. (2021) proposed a MCM that employs a neural recommender system, outperforming COMSO-RS (Song et al., 2016) and UNIFAC-IL (Song et al., 2016) on an extensive test set. Despite the promising prediction accuracies of MCMs, their applicability is inherently limited to solvents and solutes for which at least one entry in the matrix is available.

Transformers, a ML method arising from the field of natural language processing, cf. Vaswani et al. (2017), Devlin et al. (2018), have also been utilized for molecular applications, e.g., (Schwaller et al., 2019; Rong et al., 2020). Taking a sequence as input, for molecules typically consisting of SMILES strings (Weininger, 1988), transformers apply several feedforward neural network layers and attention mechanisms to learn relations within the input sequence relevant for mapping it to a property of interest, cf. Rong et al. (2020), Winter et al. (2022). Very recently, Winter et al. (2022) proposed a transformer model to predict infinite dilution ACs of solutes in solvents at varying temperature based on SMILES strings, called SMILES-to-Properties-Transformer (SPT). Yet, transformers typically require large amounts of training data in the order of millions which are not available for many molecular properties. Therefore, Winter et al. first generated about 10 million AC data points by means of COSMO-RS to pretrain the SPT and then used about 21,000 experimental AC values to fine-tune their model, thereby reaching high accuracy (Winter et al., 2022). Generation of synthetic property data for this type of pretraining is, however, computationally expensive and limited by the availability and accuracy of existing models, which can thus hinder rapid development and extension of transformers for molecular property prediction.

GNNs, another ML approach gaining great interest for molecular property prediction (Gilmer et al., 2017), have only been applied to AC prediction very recently (Sanchez Medina et al., 2022; Felton et al., 2022). GNNs operate on graph-structured data; by representing molecules as graphs with atoms as nodes and bonds as edges, GNNs can be utilized to learn molecular properties directly from molecular graphs. For AC prediction with GNNs, the solvents and solutes are thus represented as molecular graphs. Since the molecular graph representation is generally applicable to molecules, GNNs can also process solutes and solvents that have not been included in the model training. Thus, in contrast to MCMs, GNNs provide the possibility of enabling AC prediction for mixtures consisting of solutes and solvents not seen during model training. Moreover, GNNs also enable data-scarce applications without pretraining, as required for transformers, see, e.g., our previous work (Schweidtmann et al., 2020). Sanchez Medina et al. (2022) have recently utilized GNNs for the prediction of infinite dilution ACs at constant temperature for binary mixtures, achieving high prediction accuracies. They also combined GNNs with classical AC prediction models such as COSMO-RS and UNIFAC into hybrid models, where a GNN learns to correct the error of a classical model (Sanchez Medina et al., 2022). Furthermore, Felton et al. (2022) have presented DeepGamma, a deep learning model employing GNNs trained on millions of data points from COSMO-RS calculations, for predicting temperature-dependent ACs of binary mixtures at finite dilution. Also, Qin et al. (2022) have utilized COSMO-RS calculations for binary and additionally ternary solvent mixtures to develop and train SolvGNN, a GNN explicitly modeling intermolecular interactions for predicting ACs of multi-component solvent mixtures of different compositions at constant temperature. GNNs have thus already made promising advances in the field of AC prediction. However, the application of GNNs to IL solutions, the prediction of experimentally validated temperature-dependent infinite dilution ACs with GNNs, and the comparison to MCMs have not been investigated up to now.

We present a GNN for predicting temperature-dependent infinite dilution ACs of solutes in ILs.[1] In contrast to standard solvents, binary ILs constitute two disconnected but highly-attracted ionized molecules. Therefore, we develop a GNN approach that takes three molecular graphs as input (the two IL molecules and the solute), and then learns a single continuous vector representation of the IL solution, referred to as molecular fingerprint. The molecular fingerprint of the IL solution is then augmented with the temperature and mapped to the infinite dilution AC. Thereby, the GNN architecture enables an end-to-end prediction from molecular graphs of IL solutions to ACs. We thus extend current state-of-the-art GNN-based AC prediction models (Sanchez Medina et al., 2022; Felton et al., 2022; Qin et al., 2022) to IL solutions and temperature-dependent infinite dilution ACs. We analyze the GNN prediction accuracy and compare it to state-of-the-art MCM methods (Chen et al., 2021) for predicting the infinite dilution ACs of solutes in ILs. In addition, we investigate the generalization capabilities of our GNN, i.e., we analyze if the GNN is able to predict experimentally validated ACs of solutions involving ILs and solutes that were not included in the training data set.

This work is structured as follows: First, we describe the IL-solute data set (Section 2) and present the GNN model for predicting the temperature-dependent AC of solutes in ILs (Section 3). We then present and discuss the prediction performance (Section 4.1) and generalization capability (Section 4.2). Finally, we conclude our work and briefly discuss possible extensions of the presented GNN approach (Section 5).

2. Data set

We use the data set of infinite dilution ACs at varying temperatures for IL solutions that was collected from the public ILThermo database (Kazakov et al., 2013) by Chen et al. (2021). A detailed overview of the data can be found at Rittig et al. (2022a). The data set includes ILs and solutes with atoms of the types C, O, N, P, S, B, and

[1] Data, code, and trained models are openly available at Rittig et al. (2022a), where we also provide instructions on making predictions for custom data.

Table 1
Data set of IL solutions from Chen et al. (2021) categorized by molecular classes of solutes. The number of data points per solute class are shown for the total data set, the training/validation, and the test set for two different model evaluation approaches (prediction and generalization).

Solute family	All	Prediction		Generalization	
		Train/Val	Test	Train/Val	Test
Cl, F compounds	1110	1017	93	698	412
Acetic acid	29	24	5	29	0
Acetonitrile	659	563	96	634	25
Alcohols	5681	5051	630	5480	201
Aldehydes	506	489	17	455	51
Alkanes	6722	6065	657	5980	742
Alkenes	3824	3383	441	3085	739
Alkynes	2526	2275	251	2400	126
Aromatics	6550	5996	554	4634	1916
Cycloalkanes	3600	3169	431	3449	151
Esters	1446	1284	162	1368	78
Ethers	4236	3872	364	3667	569
Ketones	2265	1979	286	2154	111
Nitro alkanes	638	603	35	606	32
Pyridine	429	398	31	404	25
Terpenoids	46	46	0	46	0
Thiophene	672	600	72	647	25
Triethylamine	97	85	12	95	2
Water	517	488	29	496	21
Total	**41553**	**37387**	**4166**	**36327**	**5226**

halogens, whereas the charge of the ions within the ILs are ± 1 and each IL solution has at least ten data points, i.e., AC measurement at ten different temperatures, cf. Chen et al. (2021). The data set contains 215 ILs (consisting of 96 cations and 38 anions) and 112 solutes with a total number of 41,553 experimental γ^{∞} data points.

We provide an overview of the data set categorized by solute classes in Table 1. We divide the data set into a training/validation set, which is used for model development, and a test set, which is used after model training for comparing the model predictions against available experimental data, thereby providing an estimate of the model's prediction quality. For details on best practices for performance evaluation of data-driven models, please refer to reviews, e.g., (Gramatica, 2007; Tropsha, 2010). The number of data points we use for model training/validation and testing is shown in Table 1 with respect to two objectives: First, testing the prediction accuracy for IL solutions that contain molecules used for model training but in other combinations, and second, evaluating the prediction quality for generalization to IL solutions that contain molecules not used for model training.

For testing the prediction accuracy of our model, we use the same test set (about 10 % of the total data set) as Chen et al. (2021) who ensured that IL-solute combinations that are included in the training/validation data set are not part of the test data set. Note that we assume 10 %, i.e., more than 4000 data points, is sufficient to test the prediction quality of a model while retaining most of the valuable experimental data for training. We apply a 90 %/10 % random split to the training/validation data set.

For evaluating the generalization capability to molecular structures not seen during training, we perform another split of the whole data set into training/validation and test set, with all IL-solute combinations in the test set including at least one molecule not included in the training/validation set. Specifically, for the test set, we randomly select 5 % from all unique SMILES (Weininger, 1988) of both ILs and solutes (about 13 % of all data points); the remaining data points are used as the training/validation set. Analogously to the test set, we randomly select 5 % of unique molecules from the training/validation set to create the validation set. Each IL solution in the validation set and in the test set therefore contains at least one molecule not included in the training set and also not included in the test set or validation set, respectively. Note that for each random training/validation split in the generalization analysis, the number of data points in the validation set typically varies because both the number of IL solutions a molecule is involved in and the corresponding number of temperature-dependent ACs may vary for different molecules.

Table 2
Atom features used for initializing node attributes. All features are implemented as one-hot-encoding.

Feature	Description	Dimension
Atom type	type of atom (C, O, N, F, S, Cl, P, B, Br)	9
Is in ring	whether the atom is part of a ring	1
Is aromatic	whether the atom is part of an aromatic system	1
Charge	formal charge of the atom (−1, 0, 1)	3
Hybridization	sp, sp2, sp3, or sp3d2	4
# Hs	number of bonded hydrogen atoms	4

Table 3
Bond features used for initializing edge attributes. All features are implemented as one-hot-encoding.

Feature	Description	Dimension
Bond type	single, double, triple, or aromatic	4
Conjugated	whether the bond is conjugated	1
Is in ring	whether the bond is part of a ring	1

3. Methods & modeling

In this section, we first give a brief background on molecular graphs and GNNs, and then present our GNN model for AC prediction of solutes in ILs. We also provide insights on the MCM method we use for comparison, as well as on the training, implementation, and hyperparameter selection for both the GNN and the MCM model.

3.1. Molecular graph

GNNs take molecular graphs as input. Molecular graphs represent atoms of molecules with nodes/vertices and bonds between atoms with edges. We denote nodes (vertices) with $v \in V$ and edges connecting two nodes $v, w \in V$ with e_{vw}. In addition, each node and edge is assigned a feature vector that stores specific atom and bond information, respectively. The node feature vector is denoted by $\mathbf{f}^V(v)$ and contains, for example, information about the atom type or the formal charge of the atom. Analogously, the edge feature vector is denoted by $\mathbf{f}^E(e_{vw})$ and typically includes information about the bond type. The set of nodes and edges with the corresponding feature vectors describes the attributed molecular graph $G(m) = \{V, E, \mathbf{f}^V, \mathbf{f}^E\}$ for a molecule m.

We use the atom features shown in Table 2 and the edge features illustrated in Table 3; both are based on features reported in the literature (Gilmer et al., 2017) and our previous work on GNNs (Schweidtmann et al., 2020). For the atom features, we additionally include the formal charge since ionic liquids are composed of ionized atoms or molecules. Note that hydrogen atoms are not represented as nodes but are treated implicitly as atom feature by means of the count of hydrogen atoms bonded to a heavy atom.

3.2. Graph neural networks

GNNs (Gori et al., 2005; Scarselli et al., 2009) have recently been widely applied for molecular property prediction, e.g., in Sanchez Medina et al. (2022), Gilmer et al. (2017), Schweidtmann et al. (2020), Coley et al. (2017), Kearnes et al. (2016). GNNs learn to map the molecular graph to a property of interest in an end-to-end supervised training. We show the structure of the GNN we use for predicting the AC at varying temperatures for ionic liquids in Fig. 1. The graph-to-property structure of GNNs is based on two phases: a message passing phase and a readout phase (Gilmer et al., 2017).

In the *message passing* phase, structure information within the molecular graph is encoded by means of graph convolutions. Graph

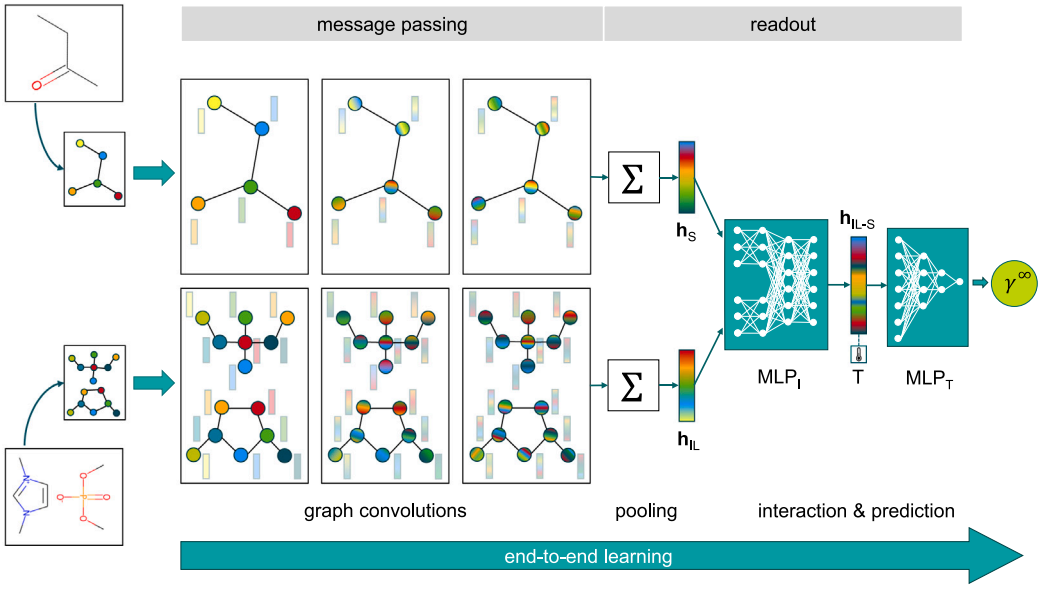

Fig. 1. Graph neural network model for prediction of the temperature-dependent infinite dilution activity coefficient of solutes in ionic liquids.

convolutions are neural network layers that operate on the attributed molecular graph (cf. Section 3.1). Specifically, the feature vector of each node v within the attributed molecular graph is considered individually and iteratively updated based on the feature vectors of the nodes and edges in the neighborhood, $N = \{w \mid e_{vw} \in E, w \neq v\}$. The updated feature vector of a node v after passing a graph convolutional layer l is typically referred to as hidden state $\mathbf{h}_v^l$. Note that the atom features are utilized to initialize the hidden state, i.e., $\mathbf{h}_v^0 = \mathbf{f}^V(v)$. The update process of a node hidden state within a graph convolution layer l can be depicted as message information from the neighborhood passed to a node v and is denoted by

$$\mathbf{h}_v^l = \mathbf{U}_l(\mathbf{h}_v^{l-1}, \sum_{w \in N(v)} \mathbf{M}_l\left(\mathbf{h}_w^{l-1}, \mathbf{f}^E\left(e_{vw}\right)\right)),$$

where a message function $\mathbf{M}_l$ produces a message from the hidden state of a neighbor node w of the previous graph convolutional layer, $\mathbf{h}_w^{l-1}$, and from the corresponding edge feature vector, $\mathbf{f}^E(e_{vw})$; then, the sum of all messages are passed to the node and are combined with the hidden state vector of the node from previous graph convolutional layer, $\mathbf{h}_w^{l-1}$, by applying an update function $\mathbf{U}_l$ which results in the updated hidden state of the node, $\mathbf{h}_v^l$. For both the message and the update function multiple variations have been proposed, e.g., GCN (Hamilton et al., 2017), GIN (Xu et al., 2018), higher-order methods (Morris et al., 2019; Flam-Shepherd et al., 2021), and approaches including 3D information of molecules like atom distances and bond angles (Schütt et al., 2018; Unke and Meuwly, 2019; Klicpera et al., 2020; Zhang et al., 2020). By stacking multiple graph convolutional layers together, each node receives information from its neighbors, with each additional layer increasing the local neighborhood information passed to a node by one additional hop (from node to node) along an edge. For example, in the second layer, the neighbors of a specific node have already received information from their respective neighbors in the previous layer which they can then pass on. The total number of graph convolutional layers L employed thus describes the local information radius, the L-hop environment of an individual node encoded in the message passing phase.

In the readout phase, the local structure information of the individual nodes is then aggregated into a continuous vector representation of the graph, the molecular fingerprint. This aggregation of the single node hidden states is conducted by means of a pooling function, e.g., the sum, $\mathbf{h}_{FP} = \sum_{v \in V} \mathbf{h}_v^L$. Taking the molecular fingerprint as an input, a feedforward neural network, typically a multilayer perceptron (MLP), is applied to predict a molecular property of interest, $\hat{p} = \text{MLP}(\mathbf{h}_{FP})$.

3.3. Graph neural network for activity coefficient prediction

Our GNN model for predicting infinite dilution ACs of solutes in ILs at varying temperatures is illustrated in Fig. 1. We first convert the molecules of an IL solution to attributed molecular graphs (cf. Section 3.1) that serve as input to the GNN. In the message passing phase of the GNN, we use two separate graph convolutional layer channels, one for the molecular graph of the solute and the other one for the two molecular graphs, i.e., anion and cation, of the IL. Thus, for the IL, the same graph convolutional layers are applied independently to the molecular graph of the anion and the molecular graph of the cation but the resulting hidden node state vectors are pooled into one IL fingerprint vector. For the graph convolutions, we apply a gated recurrent unit (GRU) with the GINE-operator (Xu et al., 2018; Hu et al., 2019) that utilizes an MLP$_{\text{GINE}}$ to map the ϵ-scaled hidden state of a node (ϵ being a learnable parameter) and the received information from the neighborhood (transformed by an activation function σ) to the updated hidden state, leading to the following update function:

$$\mathbf{h}_v^l = \text{GRU}\left(\mathbf{h}_v^{l-1}, \sigma\left(\text{MLP}_{\text{GINE}}\left((1+\epsilon)\cdot \mathbf{h}_v^{l-1} + \sum_{w \in \mathcal{N}(v)} \sigma(\mathbf{h}_w^{l-1} + \mathbf{f}_{e_{vw}})\right)\right)\right)$$

Note that here both the initial hidden node states and the edge features are linearly transformed by a learnable parameter matrix (θ) to match the dimension of the following hidden states, i.e., $\mathbf{h}_v^0 = \theta_V \cdot \mathbf{f}^V(v)$ and $\mathbf{f}_{e_{vw}} = \theta_E \cdot \mathbf{f}^E(e_{vw})$, respectively.

After the GC layers, sum pooling is applied yielding the molecular fingerprint of the IL, $\mathbf{h}_{FP,IL}$, and the solute, $\mathbf{h}_{FP,S}$, respectively. Then, the interactions between the IL and solute molecules are modeled with MLP$_I$, a MLP that transforms and concatenates the two molecular fingerprints. The output of this interaction MLP is a combined IL-solute vector $\mathbf{h}_{IL\text{-}S}$, which is then concatenated with the min–max-normalized temperature T_{norm} of the IL solution and subsequently fed into MLP$_T$, an MLP providing a prediction for the logarithmic value of the AC γ^∞, i.e.,

$$\ln(\gamma^\infty) = \text{MLP}_T([\mathbf{h}_{IL\text{-}S}^T, T_{norm}]^T).$$

3.4. Matrix completion method baseline

We re-implement an MCM model that has recently been presented for AC prediction of solutes in ILs by Chen et al. (2021), as a state-of-the-art ML-based AC prediction benchmark for our GNN model. Instead

of molecular graphs, the MCM model takes one-hot encodings of ILs and solutes as input, i.e., each IL and solute is assigned an unique ID that is encoded in a one-hot-vector. Additionally an unique ID for cation, anion, cationic family, and solute family (all four encoded as one-hot-vectors) are provided as input. For each of the six one-hot-encoded inputs, the MCM model employs several neural network layers, i.e., MLPs, to learn a continuous vector representation. The six continuous vectors are then concatenated to form a single IL solution vector. Analogously to the GNN model, the IL solution vector is then combined with the min–max-normalized temperature and mapped to $\ln(\gamma^{\infty})$ by another MLP.

3.5. Ensemble learning

For both the GNN and the MCM model we apply ensemble learning, a concept from machine learning that builds on the idea of averaging predictions of multiple models trained on different subsets of the training data set (Breiman, 1996b,a; Dietterich, 2000). Ensembles can increase the robustness of prediction models, e.g., by averaging out under- and over-predictions (Breiman, 1996a; Dietterich, 2000).

We apply ensemble learning by training multiple models on different splits of the IL-solute AC data set (cf. Section 2). Specifically, we split the data not used for testing randomly into a training and validation set before the training of each model. After training, the outputs of all models are averaged to obtain the reported AC prediction.

3.6. Implementation & hyperparameters

We implement our models in Python and utilize the geometric deep learning package PyTorch Geometric (PyG) developed by Fey & Lenssen (Fey and Lenssen, 2019). The annotated molecular graphs are generated with RDKit (Landrum, 2022). We provide the code and data used for training and testing open source, see Rittig et al. (2022a).

The proposed GNN model exhibits several hyperparameters, which we tune in a two-step process. In the first step, a grid search for the hyperparameters determining the GNN architecture is performed, varying the following hyperparameters within the respective ranges: Graph convolutional type $\in$ {NNConv, GINEConv}, number of graph convolutional layers $\in$ {1, 2, 3}, usage of GRU in graph convolutions $\in$ {True, False}, dimension of molecular fingerprint $\in$ {64, 128}, number of layers in MLP-channels in interaction network $\in$ {1, 2, 3}, activation function $\in$ {Leaky ReLU, ReLU}. Note that the number of neurons for the interaction MLP_I is not varied and is set to 256 for all MLP-channel layers except for the last MLP-channel layer that has 128 neurons, followed by three interaction layers with dimension 256. The structure of the MLP_T is not varied and set to three layers with 257 (one additional dimension for the normalized temperature), 128, and 1 neurons. The following training hyperparameters are applied: initial learning rate 0.001, learning rate decay of 0.8 with a patience of 3 epochs, batch size 64, maximum number of epochs 300, optimizer *adam*, early stopping patience of 25 epochs, dropout rate in both MLPs of 0.05. The first step of the hyperparameter search results in a final model architecture with the graph convolutional type GINEConv employed in two layers in combination with a GRU, a fingerprint dimension of 64, a number of layers in MLP-channels of 3, and Leaky RELU as activation function. In the second step of the hyperparameter tuning, a grid search to fine-tune the GNN training parameters is conducted, i.e., varying the initial learning rate $\in$ {0.01, 0.001, 0.0001}, the batch size $\in$ {32, 64, 128 }, and the dropout rate $\in$ {0.1, 0.05, 0}. We select the best model based on the validation error with a random split of the initial data set into training and validation sets. This leads to an optimal initial learning rate of 0.001, a batch size of 64, and a dropout rate of 0.

Our MCM implementation uses the model structure with MLP blocks and the hyperparameter values proposed by Chen et al. (2021). For training the MCM, we use a learning rate decay of 0.8 with a patience of 3 epochs and apply early stopping with a maximum of 300 epochs instead of a fixed number of 40 training epochs because we find the validation error to decrease in later epochs.

For choosing the ensemble size, we train 40 GNN and 40 MCM models and analyze the decrease in the validation MAE when step-wise combining more and more models of the same type starting from only 2 models. We find that the validation MAE tends to stabilize between 30 to 40 models in case of both GNN and MCM and therefore consistently use ensembles of size 40 to generate the results.

4. Results & discussion

We first evaluate AC prediction for IL solutions, where both IL and solute were in the training set, however, in different combinations than in the test set (Section 4.1). Whereas the application of MCMs is inherently limited to this application scenario, GNNs can learn molecular features from molecular graphs and thus can be applied to molecules not included in the training set. The latter application scenario is referred to as generalization, which we will investigate separately (Section 4.2).

4.1. Prediction of new IL-solute combinations

Table 4 shows the accuracies for predicting temperature-dependent ACs of new IL-solute combinations (both $\ln(\gamma^{\infty})$ and γ^{∞}). For example, the logarithmic form is relevant for calculating the chemical potential, whereas γ^{∞} is required when estimating vapor–liquid equilibrium. Note that the logarithmic form is used for model training because it exhibits a more bell-shaped distribution and avoids variations in the order of magnitude as in the unscaled AC values. The single model statistics present the accuracies on the training, the validation, and the test data averaged over 40 individual models of the same type. For the ensemble of models, distinguishing training and validation accuracies is no longer possible (cf. Section 3.5), hence we provide the accuracies for training and validation data in one category.

Table 4
Model prediction accuracies for new IL-solute combinations. Accuracy is provided by mean absolute error (MAE), root mean squared error (RMSE), coefficient of determination (R^2), and mean absolute percentage error (MAPE). For the single models, the standard deviation (±) across 40 different models is given.

	Model setup	$\ln(\gamma^{\infty})$			γ^{∞}			
		MAE	RMSE	R^2	MAE	RMSE	R^2	MAPE
GNN	single (train)	0.027 ± 0.012	0.069 ± 0.015	1.00 ± 0.00	1.4 ± 0.7	16.7 ± 9.7	0.99 ± 0.02	2.74 ± 1.19
	single (val)	0.044 ± 0.009	0.109 ± 0.013	1.00 ± 0.00	2.0 ± 0.5	18.0 ± 6.9	0.99 ± 0.01	4.67 ± 1.03
	single (test)	0.093 ± 0.004	0.162 ± 0.006	0.99 ± 0.00	7.3 ± 1.4	67.1 ± 32.2	0.79 ± 0.28	9.59 ± 0.44
	ensemble (train/val)	0.021	0.067	1.00	1.2	14.9	0.99	2.18
	ensemble (test)	0.071	0.138	0.99	6.1	51.1	0.90	7.32
MCM	single (train)	0.038 ± 0.002	0.087 ± 0.004	1.00 ± 0.00	2.9 ± 0.4	34.0 ± 4.9	0.96 ± 0.01	3.92 ± 0.23
	single (val)	0.050 ± 0.002	0.115 ± 0.009	1.00 ± 0.00	3.3 ± 0.7	32.5 ± 19.9	0.96 ± 0.04	5.30 ± 0.36
	single (test)	0.092 ± 0.002	0.157 ± 0.005	0.99 ± 0.00	5.8 ± 0.5	35.7 ± 4.2	0.95 ± 0.01	9.44 ± 0.29
	ensemble (train/val)	0.030	0.084	1.00	2.72	33.5	0.96	3.20
	ensemble (test)	0.076	0.138	0.99	5.0	30.7	0.96	7.70

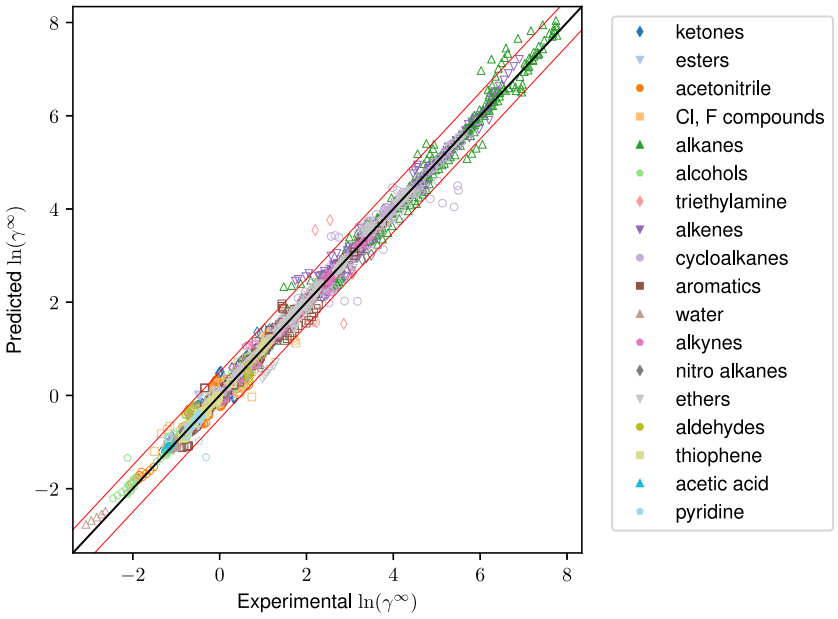

Fig. 2. GNN model ensembling parity plot for test set with new IL-solute combinations. Red lines indicate ± 0.5 error range.

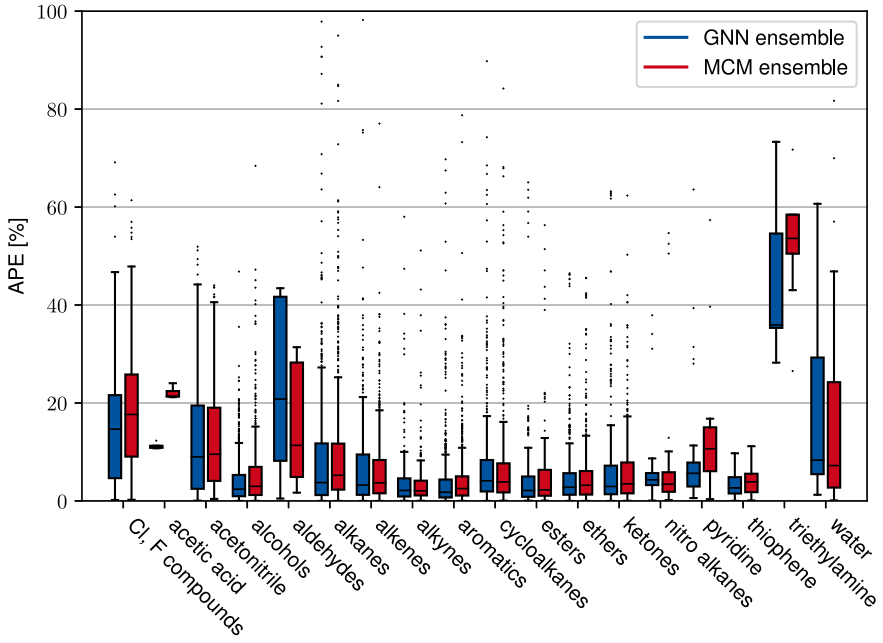

Fig. 3. Absolute percentage error (APE) of GNN ensemble and MCM ensemble for predicting γ^∞ of IL solutions in the test set categorized by solute families. For visualization, 13 outliers for the GNN ensemble and 12 outliers for the MCM ensemble with MAPE higher than 100 % are not shown.

The single GNN models achieve a mean absolute error (MAE) of 0.093 and R^2 of 0.99 on average for predicting $\ln(\gamma^\infty)$ of the IL solutions in the test set. For predicting γ^∞, the MAE amounts to 7.3 and the mean absolute percentage error is 9.6%. In case of γ^∞, the average R^2 value is visibly lower with a value of 0.79 which is caused by one of the GNN models failing to predict high γ^∞ values, also causing the high standard deviation. By using an ensemble of the single GNN models, i.e., averaging the predictions of all 40 models, the test set prediction accuracy further increases, yielding a reduced MAE of 0.071 for predicting $\ln(\gamma^\infty)$. Also, the MAE and mean absolute percentage error (MAPE) for predicting γ^∞ are reduced to values of 6.1 and 7.3%, respectively. Thus, we find an overall high prediction quality of the GNN, further enhanced by ensemble learning.

In Fig. 2, we show the parity plot of experimental and predicted logarithmic ACs for the IL solutions of the test set when using the GNN ensemble. A deviation of ±0.5 on the logarithmic scale is indicated by the red lines. We find that the $\ln(\gamma^\infty)$ predictions for almost all data points (i.e., 4109 out of 4166) are located within the ±0.5 error range. The remaining data points have mostly slightly larger errors, with the highest absolute error being 1.357. Furthermore, we do not find a systematic error.

The comparison to the state-of-the-art MCM (Chen et al., 2021) (cf. Section 3.4) shows very similar performances on the same test set are achieved by both approaches, the ensemble of GNNs and the ensemble of MCM models (cf. Table 4). Note that the implemented MCM model has a validation root mean squared error (RMSE) of 0.115 averaged

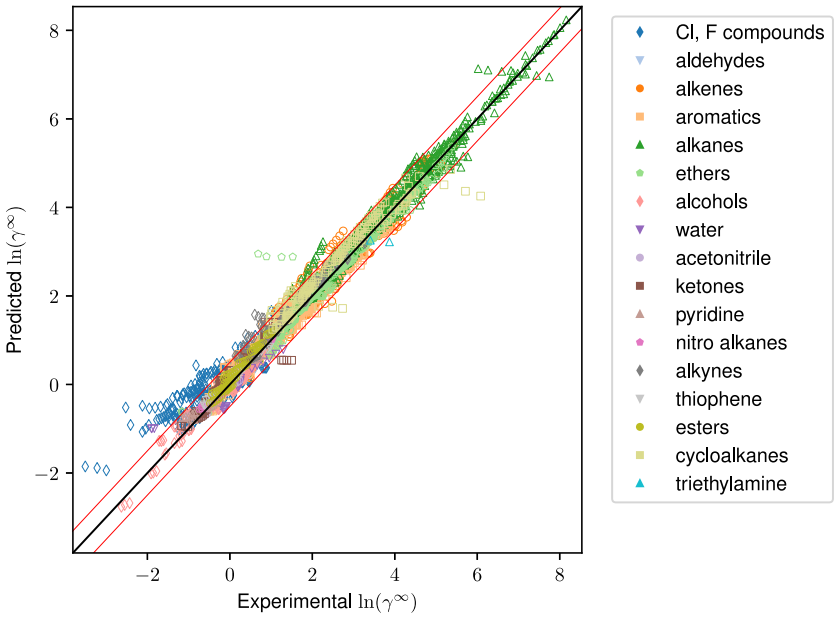

Fig. 4. GNN model ensembling parity plot for generalization test set. Red lines indicate ± 0.5 error range.

Table 5
Model prediction accuracies for generalization to IL solutions containing at least one molecule not included in the training/validation set. Accuracy is provided by mean absolute error (MAE), root mean squared error (RMSE), coefficient of determination (R^2), and mean absolute percentage error (MAPE). For the single models, the standard deviation (±) across 40 different models is given.

	Model setup	$\ln(\gamma^\infty)$			γ^∞			
		MAE	RMSE	R^2	MAE	RMSE	R^2	MAPE
GNN	single (train)	0.103 ± 0.043	0.174 ± 0.060	0.99 ± 0.01	6.0 ± 2.2	64.5 ± 19.8	0.85 ± 0.08	10.7 ± 4.5
	single (validation)	0.220 ± 0.053	0.357 ± 0.087	0.95 ± 0.03	9.6 ± 6.3	71.5 ± 58.2	0.74 ± 0.25	24.4 ± 6.1
	single (test)	0.205 ± 0.030	0.306 ± 0.052	0.96 ± 0.01	6.4 ± 1.6	48.6 ± 26.1	0.82 ± 0.30	25.1 ± 5.4
	ensemble (train/validation)	0.077	0.145	0.99	4.5	62.2	0.87	7.8
	ensemble (test)	0.156	0.230	0.98	4.0	29.4	0.95	18.0

over 40 runs each with a randomly selected validation set which is similar to the 10-fold cross-validation RMSE of 0.136 reported by Chen et al. (2021) for their best model. Also for the MCM, the application of ensemble learning considerably increases the prediction accuracy on the test set.

Chen et al. (2021) have shown that MCMs outperform classical AC prediction approaches such as the UNIFAC-IL model (Song et al., 2016) and the calibrated COSMO-RS model (Song et al., 2016). The GNN achieving competitive prediction accuracy to the MCM makes GNNs another promising alternative to UNIFAC-IL (Song et al., 2016) and the calibrated COSMO-RS (Song et al., 2016).

Finally, we show the absolute percentage error (APE) of the GNN ensemble and the MCM ensemble for predicting γ^∞ of IL solutions categorized by solute families in Fig. 3. For both the GNN ensemble and the MCM ensemble, the median APE lies below 20 % for most solute families emphasizing the high prediction quality of both models. A notable exception are solutions with triethylamine solutes with median APEs of 35.9% (MAPE of 77.3%) and 53.6% (MAPE of 71.7%) for the GNN ensemble and the MCM ensemble, respectively. We explain the high error by the low number of data points for solutions containing triethylamines and the relatively low number of molecules with nitrogen atoms across all solute molecules in the data set compared to (oxygenated) hydrocarbons. In addition, we observe outliers for many solute families (cf. Fig. 3) which indicates that despite the low MAPEs both ML models generate strong mispredictions for a small fraction of IL solutions. More than 90 % of all γ^∞ values in the test, however, are predicted with an APE below 20 % by both ensemble models.

4.2. Generalization to new IL and solute molecules

We now present the results for predicting IL solutions that contain molecules not included in the training. Table 5 shows the accuracies for predicting temperature-dependent ACs (both $\ln(\gamma^\infty)$ and γ^∞) categorized according to training, validation, and test data averaged over 40 single GNN models, as well as the ensemble learning results. Again the logarithmic form is used for model training and for the ensemble model the training and validation error is aggregated into one category (cf. Section 3.5). Comparing the prediction quality of the single GNN models to the ensemble, we again observe an accuracy increase for both training/validation and test set. In the following, we thus focus on the ensemble results.

For predicting $\ln(\gamma^\infty)$ of the IL solutions in the test set, the MAE of the ensemble amounts to 0.156 and the R^2 has a value of 0.98, indicating a high prediction quality. The MAE value for γ^∞ with 4.0 and the R^2 of 0.95 also correspond to high prediction accuracy.

Comparing the model for IL solutions with unseen molecules to the model for IL solutions with seen molecules (cf. Section 4.1), the MAE for $\ln(\gamma^\infty)$ increases for both the single GNN models and the ensemble. That increase is not surprising since generalization to new molecules is considered inherently more difficult than predicting the AC for molecules already seen during training. Since the respective test sets contain different data points, a direct quantitative comparison of the prediction accuracies, however, is not possible. Overall, a high prediction quality is maintained for the generalization to unseen molecules.

We illustrate the experimental and predicted logarithmic ACs by the ensemble for the IL solutions within the test set for generalization by

the parity plot in Fig. 4. A good match of the predicted values and the experimental values for $\ln(\gamma^{\infty})$ can be observed from the parity plot in Fig. 4. Most of the parity points lie within a deviation of ± 0.5 on the logarithmic scale indicated by the red lines (Fig. 4). For the solute class of Cl, F compounds, the predictions tend to be too high for many data points. Such deviation was not observed in the AC prediction of IL solutions with seen molecules (cf. Section 4.1), hence we do not expect the error to be inherent to the GNN model architecture or noisy data. We rather attribute the deviation to the fact that a large fraction of the Cl, F compound data points (37 %) are located in the test set (cf. Table 1) and thus fewer data points are available for training the model. For the other solute classes, we observe a low number of outliers and do not find systematic prediction errors. Thus, the parity plot in Fig. 4 emphasizes the high prediction accuracy of our GNN in case of generalization to unseen molecules.

Overall, we find that GNNs provide high prediction quality and enable generalization for AC estimation of IL solutions with unseen molecules. Predictions for molecules of classes with few data points available for training, however, should be taken with particular caution.

5. Conclusion

We present a GNN model for the prediction of temperature-dependent infinite dilution ACs of solutes in ILs. GNNs learn molecular properties based on a graph representation of molecules and have been successfully applied to AC prediction of solvents in solutes (Sanchez Medina et al., 2022; Felton et al., 2022; Qin et al., 2022). We herein extend GNNs to AC prediction of IL solutions. Specifically, we develop a GNN model that learns the infinite dilution AC as a direct function of IL and solute molecular graphs and the temperature.

The GNN model achieves high-accuracy AC predictions, superior to classical AC models such as COSMO-RS and UNIFAC-IL (Song et al., 2016) and competitive with state-of-the-art MCMs for IL solutions (Chen et al., 2021). Unlike MCMs, the GNN can also be applied to IL solutions with molecules not seen during model training, referred to as generalization. We investigate the generalization capability by excluding some molecules from training and using them for testing. Our results show that the GNN model allows for generalization with high accuracy, making it a highly promising constituent of computer-aided design of ILs.

Future work could extend GNNs for AC prediction to IL solutions at finite dilutions, similar to Felton et al. (2022). A further interesting direction is the combination of GNNs with classical AC models in form of hybrid models, cf. Sanchez Medina et al. (2022), Jirasek and Hasse (2021). Extending GNNs to provide chemically interpretable AC predictions with a quantified prediction uncertainty would also be highly desirable, with the goal of uncovering mechanistic insights that can further be used for extensions of classical AC models.

CRediT authorship contribution statement

Jan G. Rittig: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing – original draft, Visualization, Funding acquisition. **Karim Ben Hicham:** Methodology, Software, Formal analysis, Investigation, Data curation, Writing – review & editing, Visualization. **Artur M. Schweidtmann:** Conceptualization, Writing – review & editing, Funding acquisition. **Manuel Dahmen:** Conceptualization, Writing – review & editing, Supervision. **Alexander Mitsos:** Conceptualization, Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that support the findings of this study are openly available in our GitLab repository "Graph neural networks for predicting temperature-dependent activity coefficients of solutes in ionic liquids" at https://git.rwth-aachen.de/avt-svt/public/gnn_gamma_IL, cf. Rittig et al. (2022a).

Acknowledgments

This project was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 466417970 – within the Priority Programme "SPP 2331: Machine Learning in Chemical Engineering". This work was also performed as part of the Helmholtz School for Data Science in Life, Earth and Energy (HDS-LEE). Simulations were performed with computing resources granted by RWTH Aachen University under project "thes1105". AMS is supported by the TU Delft AI Labs Programme. MD received funding from the Helmholtz Association of German Research Centres.

References

Berthod, A., Ruiz-Ángel, M.J., Carda-Broch, S., 2018. Recent advances on ionic liquid uses in separation techniques. J. Chromatography A 1559, 2–16. http://dx.doi.org/10.1016/j.chroma.2017.09.044.

Breiman, L., 1996a. Bagging predictors. Mach. Learn. 24 (2), 123–140. http://dx.doi.org/10.1007/BF00058655.

Breiman, L., 1996b. Stacked regressions. Mach. Learn. 24 (1), 49–64.

Brennecke, J.F., Maginn, E.J., 2001. Ionic liquids: Innovative fluids for chemical processing. AIChE J. 47 (11), 2384–2389. http://dx.doi.org/10.1002/aic.690471102.

Butler, K.T., Davies, D.W., Cartwright, H., Isayev, O., Walsh, A., 2018. Machine learning for molecular and materials science. Nature 559 (7715), 547–555. http://dx.doi.org/10.1038/s41586-018-0337-2.

Chen, G., Song, Z., Qi, Z., Sundmacher, K., 2021. Neural recommender system for the activity coefficient prediction and UNIFAC model extension of ionic liquid–solute systems. AIChE J. 67 (4), http://dx.doi.org/10.1002/aic.17171.

Coley, C.W., Barzilay, R., Green, W.H., Jaakkola, T.S., Jensen, K.F., 2017. Convolutional embedding of attributed molecular graphs for physical property prediction. J. Chem. Inform. Model. 57 (8), 1757–1772. http://dx.doi.org/10.1021/acs.jcim.6b00601.

Damay, J., Jirasek, F., Kloft, M., Bortz, M., Hasse, H., 2021. Predicting activity coefficients at infinite dilution for varying temperatures by matrix completion. Ind. Eng. Chem. Res. 60 (40), 14564–14578. http://dx.doi.org/10.1021/acs.iecr.1c02039.

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv, arXiv preprint arXiv:1810.04805v2.

Dietterich, T.G., 2000. Ensemble methods in machine learning. In: Multiple Classifier Systems: First International Workshop. MCS 2000, In: Lecture Notes in Computer Science, 21.06.-23.06.2000, Cagliari, Italy, pp. 1–15.

Dong, Y., Huang, S., Guo, Y., Lei, Z., 2020. COSMO–UNIFAC model for ionic liquids. AIChE J. 66 (1), http://dx.doi.org/10.1002/aic.16787.

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

Fey, M., Lenssen, J.E., 2019. Fast graph representation learning with PyTorch geometric. arXiv preprint arXiv:1903.02428v3.

Flam-Shepherd, D., Wu, T.C., Friederich, P., Aspuru-Guzik, A., 2021. Neural message passing on high order paths. Mach. Learn.: Sci. Technol. 2 (4), 045009. http://dx.doi.org/10.1088/2632-2153/abf5b8.

Fredenslund, A., Jones, R.L., Prausnitz, J.M., 1975. Group-contribution estimation of activity coefficients in nonideal liquid mixtures. AIChE J. 21 (6), 1086–1099. http://dx.doi.org/10.1002/aic.690210607.

Galiński, M., Lewandowski, A., Stepniak, I., 2006. Ionic liquids as electrolytes. Electrochim. Acta 51 (26), 5567–5580. http://dx.doi.org/10.1016/j.electacta.2006.03.016.

Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E., 2017. Neural message passing for quantum chemistry. In: 34th International Conference on Machine Learning, Vol. 3. ICML 2017, pp. 2053–2070, arXiv:1704.01212.

Gori, M., Monfardini, G., Scarselli, F., 2005. A new model for earning in graph domains. In: Proceedings of the International Joint Conference on Neural Networks, Vol. 2. IEEE, pp. 729–734. http://dx.doi.org/10.1109/IJCNN.2005.1555942.

Gramatica, P., 2007. Principles of QSAR models validation: Internal and external. QSAR Combinatorial Sci. 26 (5), 694–701. http://dx.doi.org/10.1002/qsar.200610151.

Hamilton, W., Ying, Z., Leskovec, J., 2017. Inductive representation learning on large graphs. In: Advances in Neural Information Processing Systems, Vol. 30. NIPS 2017, pp. 1024–1034.

Han, X., Armstrong, D.W., 2007. Ionic liquids in separations. Acc. Chem. Res. 40 (11), 1079–1086. http://dx.doi.org/10.1021/ar700044y.

Han, J., Dai, C., Yu, G., Lei, Z., 2018. Parameterization of COSMO-RS model for ionic liquids. Green Energy Environ. 3 (3), 247–265. http://dx.doi.org/10.1016/j.gee.2018.01.001.

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.-S., 2017. Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web. WWW '17, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, pp. 173–182. http://dx.doi.org/10.1145/3038912.3052569.

Hu, W., Liu, B., Gomes, J., Zitnik, M., Liang, P., Pande, V., Leskovec, J., 2019. Strategies for pre-training graph neural networks. arXiv preprint arXiv:1905.12265v3.

Jirasek, F., Alves, R.A.S., Damay, J., Vandermeulen, R.A., Bamler, R., Bortz, M., Mandt, S., Kloft, M., Hasse, H., 2020. Machine learning in thermodynamics: Prediction of activity coefficients by matrix completion. J. Phys. Chem. Lett. 11 (3), 981–985. http://dx.doi.org/10.1021/acs.jpclett.9b03657.

Jirasek, F., Hasse, H., 2021. Machine learning of thermophysical properties. Fluid Phase Equilib. 549, 113206. http://dx.doi.org/10.1016/j.fluid.2021.113206.

Karunanithi, A.T., Mehrkesh, A., 2013. Computer-aided design of tailor-made ionic liquids. AIChE J. 59 (12), 4627–4640. http://dx.doi.org/10.1002/aic.14228.

Kazakov, A., Magee, J., Chirico, R., Diky, V., Kroenlein, K., Muzny, C., Frenkel, M., 2013. Ionic Liquids Database - ILThermo (v2.0). Ionic Liquids Database - ILThermo (v2.0).

Kearnes, S., McCloskey, K., Berndl, M., Pande, V., Riley, P., 2016. Molecular graph convolutions: Moving beyond fingerprints. J. Comput. Aided Mol. Des. 30 (8), 595–608. http://dx.doi.org/10.1007/s10822-016-9938-8.

Klamt, A., 1995. Conductor-like screening model for real solvents: A new approach to the quantitative calculation of solvation phenomena. J. Phys. Chem. 99 (7), 2224–2235. http://dx.doi.org/10.1021/j100007a062.

Klicpera, J., Groß, J., Günnemann, S., 2020. Directional message passing for molecular graphs. arXiv preprint arXiv:2003.03123.

Landrum, G., 2022. RDKit: Open-source cheminformatics software. http://www.rdkit.org. (Accessed 14 June 2022).

Lei, Z., Chen, B., Koo, Y.-M., MacFarlane, D.R., 2017. Introduction: Ionic liquids. Chem. Rev. 117 (10), 6633–6635. http://dx.doi.org/10.1021/acs.chemrev.7b00246.

Lei, Z., Dai, C., Liu, X., Xiao, L., Chen, B., 2012. Extension of the UNIFAC model for ionic liquids. Ind. Eng. Chem. Res. 51 (37), 12135–12144. http://dx.doi.org/10.1021/ie301159v.

Lei, Z., Dai, C., Zhu, J., Chen, B., 2014. Extractive distillation with ionic liquids: A review. AIChE J. 60 (9), 3312–3329. http://dx.doi.org/10.1002/aic.14537.

Lei, Z., Zhang, J., Li, Q., Chen, B., 2009. UNIFAC model for ionic liquids. Ind. Eng. Chem. Res. 48 (5), 2697–2704. http://dx.doi.org/10.1021/ie801496e.

Morris, C., Ritzert, M., Fey, M., Hamilton, W.L., Lenssen, J.E., Rattan, G., Grohe, M., 2019. Weisfeiler and Leman go neural: Higher-order graph neural networks. In: 33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019. pp. 4602–4609.

Muratov, E.N., Bajorath, J., Sheridan, R.P., Tetko, I.V., Filimonov, D., Poroikov, V., Oprea, T.I., Baskin, I.I., Varnek, A., Roitberg, A., Isayev, O., Curtarolo, S., Fourches, D., Cohen, Y., Aspuru-Guzik, A., Winkler, D.A., Agrafiotis, D., Cherkasov, A., Tropsha, A., 2020. QSAR without borders. Chem. Soc. Rev. 49 (11), 3525–3564. http://dx.doi.org/10.1039/D0CS00098A.

Peng, D., Zhang, J., Cheng, H., Chen, L., Qi, Z., 2017. Computer-aided ionic liquid design for separation processes based on group contribution method and COSMO-SAC model. Chem. Eng. Sci. 159, 58–68. http://dx.doi.org/10.1016/j.ces.2016.05.027.

Plechkova, N.V., Seddon, K.R., 2008. Applications of ionic liquids in the chemical industry. Chem. Soc. Rev. 37 (1), 123–150. http://dx.doi.org/10.1039/B006677J.

Qin, S., Jiang, S., Li, J., Balaprakash, P., van Lehn, R., Zavala, V., 2022. Capturing molecular interactions in graph neural networks: A case study in multi-component phase equilibrium. http://dx.doi.org/10.26434/chemrxiv-2022-3tq4c, ChemRxiv preprint 10.26434/chemrxiv-2022-3tq4c.

Rittig, J.G., Ben Hicham, K., Schweidtmann, A.M., Dahmen, M., Mitsos, A., 2022a. Open-source graph neural networks for predicting temperature-dependent activity coefficients of solutes in ionic liquids. https://git.rwth-aachen.de/avt-svt/public/gnn_gamma_IL. (Accessed 14 June 2022).

Rittig, J.G., Gao, Q., Dahmen, M., Mitsos, A., Schweidtmann, A.M., 2022b. Graph neural networks for the prediction of molecular structure-property relationships. arXiv preprint arXiv:2208.04852.

Rogers, R.D., Seddon, K.R., 2003. Chemistry. Ionic liquids–solvents of the future? Science 302 (5646), 792–793. http://dx.doi.org/10.1126/science.1090313.

Rong, Y., Bian, Y., Xu, T., Xie, W., WEI, Y., Huang, W., Huang, J., 2020. Self-supervised graph transformer on large-scale molecular data. In: Advances in Neural Information Processing Systems 33. NeurIPS 2020, Curran Associates, Inc., pp. 12559–12571.

Roughton, B.C., Christian, B., White, J., Camarda, K.V., Gani, R., 2012. Simultaneous design of ionic liquid entrainers and energy efficient azeotropic separation processes. Comput. Chem. Eng. 42, 248–262. http://dx.doi.org/10.1016/j.compchemeng.2012.02.021.

Sanchez Medina, E.I., Linke, S., Stoll, M., Sundmacher, K., 2022. Graph neural networks for the prediction of infinite dilution activity coefficients. Digit. Discov. 1, 216–225. http://dx.doi.org/10.1039/D1DD00037C.

Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G., 2009. Computational capabilities of graph neural networks. IEEE Trans. Neural Netw. 20 (1), 81–102. http://dx.doi.org/10.1109/TNN.2008.2005141.

Schütt, K.T., Sauceda, H.E., Kindermans, P.J., Tkatchenko, A., Müller, K.R., 2018. SchNet - A deep learning architecture for molecules and materials. J. Chem. Phys. 148 (24), 1–11. http://dx.doi.org/10.1063/1.5019779, arXiv:1712.06113.

Schwaller, P., Laino, T., Gaudin, T., Bolgar, P., Hunter, C.A., Bekas, C., Lee, A.A., 2019. Molecular transformer: A model for uncertainty-calibrated chemical reaction prediction. ACS Central Sci. 5 (9), 1572–1583. http://dx.doi.org/10.1021/acscentsci.9b00576.

Schweidtmann, A.M., Rittig, J.G., König, A., Grohe, M., Mitsos, A., Dahmen, M., 2020. Graph neural networks for prediction of fuel ignition quality. Energy Fuels 34 (9), 11395–11407. http://dx.doi.org/10.1021/acs.energyfuels.0c01533.

Seddon, K.R., 1997. Ionic liquids for clean technology. J. Chem. Technol. Biotechnol. 68 (4), 351–356. http://dx.doi.org/10.1002/(SICI)1097-4660(199704)68:4%3C351::AID-JCTB613%3E3.0.CO;2-4.

Song, Z., Li, X., Chao, H., Mo, F., Zhou, T., Cheng, H., Chen, L., Qi, Z., 2019. Computer-aided ionic liquid design for alkane/cycloalkane extractive distillation process. Green Energy Environ. 4 (2), 154–165. http://dx.doi.org/10.1016/j.gee.2018.12.001.

Song, Z., Zhang, C., Qi, Z., Zhou, T., Sundmacher, K., 2018. Computer-aided design of ionic liquids as solvents for extractive desulfurization. AIChE J. 64 (3), 1013–1025. http://dx.doi.org/10.1002/aic.15994.

Song, Z., Zhang, J., Zeng, Q., Cheng, H., Chen, L., Qi, Z., 2016. Effect of cation alkyl chain length on liquid-liquid equilibria of ionic liquids + thiophene + heptane: COSMO-RS prediction and experimental verification. Fluid Phase Equilib. 425, 244–251. http://dx.doi.org/10.1016/j.fluid.2016.06.016.

Tropsha, A., 2010. Best practices for QSAR model development, validation, and exploitation. Mol. Inform. 29 (6–7), 476–488. http://dx.doi.org/10.1002/minf.201000061.

Unke, O.T., Meuwly, M., 2019. PhysNet: A neural network for predicting energies, forces, dipole moments, and partial charges. J. Chem. Theory Comput. 15 (6), 3678–3693. http://dx.doi.org/10.1021/acs.jctc.9b00181.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need. In: Advances in Neural Information Processing Systems 30. NIPS 2017, Curran Associates, Inc., pp. 5998–6008.

Wang, J., Song, Z., Cheng, H., Chen, L., Deng, L., Qi, Z., 2018. Computer-aided design of ionic liquids as absorbent for gas separation exemplified by CO_2 capture cases. ACS Sustain. Chem. Eng. 6 (9), 12025–12035. http://dx.doi.org/10.1021/acssuschemeng.8b02321.

Weininger, D., 1988. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. J. Chem. Inform. Model. 28 (1), 31–36. http://dx.doi.org/10.1021/ci00057a005.

Winter, B., Winter, C., Schilling, J., Bardow, A., 2022. A smile is all you need: Predicting limiting activity coefficients from SMILES with natural language processing. arXiv, arXiv preprint arXiv:2206.07048.

Xu, K., Hu, W., Leskovec, J., Jegelka, S., 2018. How powerful are graph neural networks? arXiv preprint arXiv:1810.00826v3.

Zeng, S., Zhang, X., Bai, L., Zhang, X., Wang, H., Wang, J., Bao, D., Li, M., Liu, X., Zhang, S., 2017. Ionic-liquid-based CO_2 capture systems: Structure, interaction and process. Chem. Rev. 117 (14), 9625–9673. http://dx.doi.org/10.1021/acs.chemrev.7b00072.

Zhang, S., Liu, Y., Xie, L., 2020. Molecular mechanics-driven graph neural network with multiplex graph for molecular structures. arXiv preprint arXiv:2011.07457.

Zhang, Z., Song, J., Han, B., 2017. Catalytic transformation of lignocellulose into chemicals and fuel products in ionic liquids. Chem. Rev. 117 (10), 6834–6880. http://dx.doi.org/10.1021/acs.chemrev.6b00457.