

Hybrid Graph Neural Networks for the prediction of activity coefficients in separation processes

Dissertation

zur Erlangung des akademischen Grades

**Doktoringenieur
(Dr.-Ing.)**

von M.Sc. Edgar Ivan Sanchez Medina

geboren am 12.09.1995 in Mazatlan, Sinaloa, Mexiko

genehmigt durch die Fakultät für Verfahrens- und Systemtechnik der
Otto-von-Guericke-Universität Magdeburg

Promotionskommission:

Prof. Dr. Berend van Wachem (Vorsitz)

Prof. Dr.-Ing. Kai Sundmacher (Gutachter)

Prof. Dr. Martin Stoll (Gutachter)

Prof. Dr.-Ing. Hans Hasse (Gutachter)

eingereicht am: 06.09.2024

Promotionskolloquium am: 17.12.2024

M.Sc. Edgar Ivan Sanchez Medina

Hybrid Graph Neural Networks for the prediction of activity coefficients in separation processes

Dissertation, 06.09.2024

Otto-von-Guericke-Universität Magdeburg

Department of Process Systems Engineering

Institute of Process Engineering

Universitätsplatz 2

39106 Magdeburg

Abstract

The task of predicting properties of mixtures from the molecular structure of their components has been studied for decades. In the past, a broad spectrum of mechanistic models has been developed for predicting thermophysical properties of mixtures. These models have been the basis of many successful applications across various chemical and process engineering domains. Noticeable examples are the large chemical plants supporting today's world economy through oil refinement.

However, when the properties of novel complex mixtures are the focus, e.g., occurring in biorefineries or in chemical recycling of plastic waste streams, or when sustainable processes for the future circular economy need to be developed, the existing property models are often limited regarding their accuracy and predictive power and are not suited for the effective exploration of the vast chemical space.

Of special importance is the modeling and calculation of phase equilibria, which is a cornerstone in the design of separation processes for molecular mixtures. For describing the non-ideal mixing behaviour of components in liquid mixtures, activity coefficients are often used. The exploration of alternative routes to separation requires the development of accurate and efficient predictive models that estimate the activity coefficients across large chemical spaces.

In this dissertation several hybrid models are presented that combine the flexibility of graph neural networks (GNNs) with phenomenological/mechanistic modeling approaches of the thermodynamic behavior of mixtures. Two main arrangements for the construction of such hybrid models are explored: (i) a parallel arrangement in which the graph neural network serves as a corrector of a phenomenological model prediction. (ii) A serial arrangement in which the graph neural network is embedded in some form of mechanistic expression to preserve the physical constraints of the latter.

The proposed hybrid graph neural network models are then presented from the simplest scenario to increasing levels of generality in predicting activity coefficients. The proposed models are extensively tested to evaluate their advantages and limitations compared to conventional methods (e.g., UNIFAC). The dissertation concludes with a series of case studies that demonstrate the utility of the proposed models in the context of supporting the early stages of separation process design.

Overall, the results suggest that hybrid graph neural networks offer more efficient and accurate solutions for predicting activity coefficients compared to the standalone submodels. Such advantages can be exploited in practical scenarios in the context of separation process design. The implementation of hybrid graph neural networks could be of high relevance in the development of more sustainable separation processes.

Zusammenfassung

Die Vorhersage der Eigenschaften von Stoffgemischen anhand der Molekularstruktur ihrer Bestandteile wird seit mehreren Jahrzehnten erforscht. Im Ergebnis dieser Aktivitäten wurde eine ganze Reihe von mechanistischen Modellen der thermophysikalischen Eigenschaften von Reinstoffen und Gemischen erfolgreich entwickelt. Diese Modelle bildeten die Grundlage für zahlreiche erfolgreiche Anwendungen in verschiedenen Bereichen der chemischen und thermischen Verfahrenstechnik. Ein wichtiges Beispiel sind die Chemieanlagen in Erdölraffinerien, welche das Rückgrat der heutigen (fossilen) Weltwirtschaft darstellen.

Wenn jedoch die Eigenschaften neuartiger komplexer Gemische im Mittelpunkt stehen, wie sie z. B. in Bioraffinerien oder beim chemischen Recycling von Kunststoffabfallströmen vorkommen, oder wenn nachhaltige Prozesse für die zukünftige Kreislaufwirtschaft entwickelt werden sollen, dann sind die existierenden Eigenschaftsmodelle in Bezug auf ihre Genauigkeit und Vorhersagekraft oft begrenzt und eignen sich insbesondere nicht für die effektive Erkundung großer Räume von molekularen Strukturen.

Von besonderer Bedeutung für die thermische Verfahrenstechnik ist die Modellierung von Phasengleichgewichte von Gemischen. Sie ist der Grundstein für die Auslegung von Trennprozessen für Flüssigkeitsgemische. Zur Beschreibung des nicht-idealen Verhaltens von Komponenten in Flüssigkeitsgemischen werden häufig Aktivitätskoeffizienten verwendet. Um Aktivitätskoeffizienten für verschiedenste Moleküle in große chemischen Strukturräumen zuverlässig bestimmen zu können, werden akkurate, effizient auswertbare Vorhersagemodelle benötigt.

Vor diesem Hintergrund werden in der vorliegenden Dissertation verschiedene hybride Modelle entwickelt, die die Flexibilität von Graph Neural Networks (GNNs) mit mechanistischen Modellen des thermodynamischen Verhaltens von Gemischen kombinieren. Es werden zwei Hauptanordnungen für die Konstruktion solcher Hybridmodelle untersucht: (i) eine parallele Anordnung, bei der ein Graph Neural Network als Korrektor für eine mechanistischen Modellvorhersage dient; (ii) eine serielle Anordnung, bei der ein Graph Neural Network in ein mechanistisches Modell direkt eingebettet ist, um bestimmte physikalisch begründete Eigenschaften des letzteren zu erhalten.

Die vorgeschlagenen hybriden GNNs werden dann vom einfachsten Szenario bis hin zu allgemeineren Vorhersagen von Aktivitätskoeffizienten vorgestellt. Die hybriden Modelle werden ausgiebig getestet, um ihre Vorteile und Grenzen im Vergleich zu konventionellen Methoden (z. B. UNIFAC) aufzuzeigen. Die Dissertation schließt

mit einer Reihe von Fallstudien, die den Nutzen der vorgeschlagenen Modelle in der frühen Phase der Planung von Trennprozessen aufzeigen.

Insgesamt deuten die erzielten Ergebnisse darauf hin, dass hybride GNNs im Vergleich zu den eigenständigen Teilmodellen effizientere und genauere Lösungen für die Vorhersage von Aktivitätskoeffizienten liefern. Diese Vorteile können in praktischen Anwendungen im Zusammenhang mit dem Entwurf von Trennprozessen genutzt werden. Die Implementierung dieser hybriden GNN-Modelle könnte insbesondere für die Entwicklung nachhaltigerer Trennverfahren zukünftig von großer Bedeutung sein.

Preface

Writing this dissertation has been a unique opportunity for me to condense the efforts and findings of most of my time as a doctoral researcher. Along with each Chapter of this dissertation, I have included a quote. Collectively, I believe that these quotes represent the spirit of this dissertation, and I would like, in this preface, to provide the reader with a more extensive explanation of why I believe it so.

The collective capacity that humanity has achieved compared to other living species is very admirable. However, more often than not, we forget about what unites us as a species, and the struggles and wonders that we all share. As Prof. Goodenough said, “we compete against problems, not against people”. I believe that this attitude will be of great practical value in addressing today’s sustainability challenge.

This dissertation combines three main pillars of knowledge: phase equilibria, hybrid modeling, and graph neural networks. While I understand that the core advances in these areas have historically happened independent from each other, the advancement of science will increasingly require a multidisciplinary approach. Of course, this poses certain challenges, since our individual capacities are limited. Historically, these individual limitations have usually been overcome by the collective development of knowledge. This point was also considered by Max Planck, to whom the phrase “science advances one funeral at a time” is attributed.

While our goal should be to arrive at a complete understanding of physical phenomena, it is also true that practical challenges require the use of tractable solutions. This was mentioned by Arnold Bondi in the very same context of predicting properties of condensed matter from molecular structure [22]. In fact, Bondi’s work on mapping molecular structure to property prediction is still relevant and used today. This has motivated the use of graph neural networks and hybrid parallel approaches in this work.

Nevertheless, our understanding of physical phenomena is not completely missing, and the physical constraints that we do understand must always be respected. Therefore, as Helmholtz pointed out - “Each individual fact, taken by itself, can indeed arouse our curiosity or our astonishment, or be useful to us in its practical applications. But intellectual satisfaction we obtain only from a connection of the whole, just from its conformity with law”.

As the reader will soon discover, great part of the models proposed in this dissertation contain a large deep learning component. I believe to be though provoking to include the quote - "one doesn't bet against deep learning" by Ilya Sutskever precisely on Chapter 5, where several limitations of the proposed approaches are pointed out. However, while observing the limitations is important, one should not forget the significant benefits, as illustrated in this dissertation, that deep learning can provide. Such benefits can also permeate the realm of science in general, and the area of process systems engineering in particular.

In the past, Prof. Roger Sargent foresaw the importance that computers would have for process engineering. This was at a time when his own machine was extremely limited by today's standards. I think we can now confidently say that his prediction was correct. Overall, I expect deep learning-enabled approaches to be of tremendous importance in several areas of science and, specifically, process systems engineering. It is my hope that this dissertation can support these advances.

Nomenclature

Latin

a	activity
A	Porter's model parameter or KDB vapor pressure correlation coefficient
\mathbf{A}	matrix of node features
\mathbf{a}	vector of node features
AP	atomic polarizability
B	second-virial coefficient or KDB vapor pressure correlation coefficient
\mathbf{B}	matrix of edge features
\mathbf{b}	vector of edge features
BP	bond polarizability
C	KDB vapor pressure correlation coefficient
\mathbf{C}	matrix of graph connectivity
D	KDB vapor pressure correlation coefficient
f	fugacity or function
FP	molecular fingerprint
G	Gibbs energy
g	molar Gibbs energy
\bar{g}	partial molar Gibbs energy
H	enthalpy
\bar{h}	partial molar enthalpy
IG	integrated gradients
K	weighting factor or parameter from Gibbs-Helmholtz derived expression
K_α	separation factor
K_∞	separation factor at infinite dilution
L	number of message passing layers
M	system's property or molar mass
m	molar property
\bar{m}	partial molar property
$\text{min}SF$	minimum solvent-to-feed ratio
n	number of moles
\mathbf{n}	vector of number of moles
n_a	number of nodes in a graph
n_b	number of edges in a graph
n_D	number of data points
N	number of mixture components
N^{HBA}	number of hydrogen-bond acceptor sites
N^{HBD}	number of hydrogen-bond donor sites
p	partial pressure
P	pressure
\mathbf{q}	vector of learnable parameters or hidden state of LSTM
R	universal gas constant
r	residual
\mathbf{r}	readout vector of LSTM

SF	solvent-to-feed ratio
S	entropy or selectivity
T	temperature
$TopoPSA$	topological polar surface area
u	global-level feature
\mathbf{u}	vector of global-level features
v	molar volume
V	volume
\bar{v}	partial molar volume
w	Margules parameter
\mathbf{W}	matrix of learnable parameters
x	molar fraction in the liquid phase
\mathbf{x}	vector of molar fractions in the liquid phase
y	molar fraction in the vapor phase
z	molar fraction
\mathbf{z}	vector of molar fractions

Greek

α	attention weight or relative volatility
β	vector of learnable parameters
δ	Jaccard distance metric
ϵ	small number
γ	activity coefficient
μ	chemical potential
Ω	weight fraction activity coefficient
ϕ	fugacity coefficient
ϕ_0	initial features transformation function
ϕ_a	node updating function
ϕ_b	edge updating function
ϕ_E	edge feature transformation function
ϕ_u	global updating function
Π	number of phases at equilibrium
ψ	message passing function
Ξ	updating function

Subscripts

$base$	baseline
CL	caprolactam
g	graph-level
\mathcal{G}	actual graph
\mathcal{G}_{base}	baseline graph
i	mixture component i
IL	ionic-liquid
$\neq i$	different than component i
j	mixture component j
k	mixture component k
mg	mixture graph
mix	mixture
$pure$	pure chemical species

r	role of the species (either solvent or solute)
s	extrapolated chemical species
S	solvent
t	processing step of LSTM
v	node v
vw	between node v and node w
w	node w

Superscripts

0	reference state
$(final)$	final embedding
(i)	phase i
(l)	number of message passing layer
*	final state in processing step of LSTM
Cat	concatenated
E	excess property or extract phase
hyb	hybrid
id	ideal
∞	infinite dilution
L	liquid phase
phe	phenomenological
$pure$	pure component
R	raffinate phase
sat	saturation
V	vapor phase

Other symbols

A	set of pure component molecular-level characteristics
B	set of pure component phase-level properties
C	set of mixture components' molecular-level characteristics
\mathcal{CO}	set of mixture components
\mathcal{D}	set of mixture phase-level properties
\mathcal{E}	set of graph edges
\mathcal{G}	graph
\mathcal{M}	DECHEMA training set
\mathcal{N}	neighborhood
\mathcal{P}	global-pooling function
\mathcal{T}	set of 10 shortest Jaccard distances
\mathcal{V}	set of graph nodes
\square	aggregation operator

Acronyms and Abbreviations

Adam	adaptive moment estimation
AE	absolute error
CAMD	computer-aided molecular design
CAS-RN	chemical abstracts service registry number
CAMPD	computer-aided molecular and process design
COSMO	conductor-like screening model

COSMO-RS	conductor-like screening model for real solvents
DECHEMA	Deutsche Gesellschaft für Chemische Technik und Biotechnologie
DFT	density functional theory
e-GNN	ensemble of graph neural networks
GAT	graph attention network
GCN	graph convolutional network
GH-GNN	Gibbs-Helmholtz Graph Neural Network
GHS	globally harmonized system
GNN	graph neural network
GPU	graphics process unit
GRU	gated recurrent unit
HSP	Hansen solubility parameter
IDAC	infinite dilution activity coefficient
IL	ionic-liquid
KDB	Korean Data Bank
LLE	liquid-liquid equilibrium
LSTM	long-short term memory
LSER	linear solvation energy relationship
MAE	mean absolute error
MAPE	mean absolute percentage error
MCM	matrix completion method
MSE	mean squared error
MPI	Max Planck Institute
MPNN	message passing neural network
MLP	multi-layer perceptron
MOSCED	modified separation of cohesive energy density
NRTL	nonrandom two-liquid model
QSPR	quantitative structure-property relationship
ReLU	rectified linear unit
R ²	coefficient of determination
SLE	solid-liquid equilibrium
SLEL	solid-liquid-liquid equilibrium
SMILES	simplified molecular-input line-entry system
SPT	SMILES-to-property transformer
TCM	tensor completion method
UNIQUAC	universal quasichemical model
UNIFAC	universal quasichemical functional-group activity coefficients model
VLE	vapor-liquid equilibrium

Contents

Abstract	iii
Preface	vii
Nomenclature	ix
1 Introduction	1
1.1 Motivation	3
1.2 Contributions of this thesis	5
1.3 Thesis structure	6
2 Background	9
2.1 Phase equilibria	9
2.1.1 Thermodynamic fundamentals	10
2.1.2 Activity coefficients	13
2.2 Hybrid data-driven and mechanistic modeling	17
2.2.1 Parallel arrangement	18
2.2.2 Serial arrangement	19
2.3 Graph neural networks (GNNs)	20
3 Predicting Isothermal Infinite Dilution Activity Coefficients	23
3.1 Data sources and data cleaning	25
3.2 Molecular graphs	27
3.3 Graph neural network	29
3.3.1 Training	34
3.3.2 Ensemble learning	34
3.3.3 Comparison to phenomenological models	35
3.3.4 Robustness analysis with 5-fold cross-validation	39
3.4 Hybrid parallel graph neural networks	41
3.4.1 Hybrid UNIFAC-based GNN models	43
3.4.2 Hybrid COSMO-RS GNN model	45
3.4.3 Hybrid Abraham and MOSCED GNN models	47

3.5	Additional isothermal studies	48
3.5.1	Data preprocessing	48
3.5.2	Model comparison	51
3.6	Chapter summary	57
4	Predicting Temperature-Dependent Infinite Dilution Activity Coefficients	59
4.1	Gibbs-Helmholtz Graph Neural Network (GH-GNN)	60
4.1.1	Data set	61
4.1.2	Molecular graphs with global features	62
4.1.3	Model architecture	64
4.1.4	Multi-task pre-training	70
4.1.5	Model performance	70
4.2	Extending the GH-GNN model to ionic liquids	82
4.2.1	Data set	82
4.2.2	Extension strategies	82
4.3	Extending the GH-GNN model to polymer solutions	85
4.3.1	Data set	87
4.3.2	Polymer representations	88
4.3.3	Interpolating among systems	88
4.3.4	Extrapolating to other small molecules	91
4.4	Chapter summary	92
5	Predicting Activity Coefficients	95
5.1	From infinite to finite using the Margules equation	96
5.2	Predicting binary vapor-liquid equilibria	97
5.2.1	Data set	98
5.2.2	Isothermal vapor-liquid equilibria	100
5.2.3	Isobaric vapor-liquid equilibria	104
5.2.4	Overall performance on binary vapor-liquid equilibria	108
5.3	Predicting ternary vapor-liquid equilibria	110
5.4	Chapter summary	113
6	Applications to Separation Processes	115
6.1	Pre-selecting solvents for extractive distillation	115
6.1.1	Selection based on selectivity at infinite dilution	117
6.1.2	Selection based on relative volatility at infinite dilution	118
6.1.3	Selection based on minimum solvent-to-feed ratio	119
6.1.4	Results of solvent pre-selection	121
6.2	Pre-selecting solvents for liquid-liquid extraction of caprolactam from ionic-liquid	127

6.3	Graph neural networks assisting the design of a lignin fractionation process	131
6.4	Chapter summary	138
7	Conclusions	141
7.1	Summary	141
7.2	Outlook	143
	Bibliography	145
A	Appendices	167
A.1	Relationship between partial molar properties and state variables . .	167
A.2	Chemical potential in terms of fugacity	169
A.3	Derivation of the dependency expressions of γ_i on T , P and \mathbf{x}	172
A.4	Derivation of the relationship of excess chemical potential and γ_i . .	175
A.5	Solvation models: Hildebrand parameter, Hansen Solubility Parameters and MOSCED	176
A.6	Group contribution models: UNIFAC, UNIFAC-Lyngby and UNIFAC-Dortmund	178
A.7	Abraham model	181
A.8	Hyperparameters for the isothermal-IDACs GNN model	182
A.9	Distribution of IDAC values in the isothermal 298.15 K data set . . .	183
A.10	Errors in the Vol. IX of DECHEMA Chemistry Data Series	184
A.11	Details of the DECHEMA data set of IDACs	186
A.12	Hyperparameters for e-GNNprev and e-SolvGNN in extended isothermal analysis	190
A.13	Hyperparameters for the GH-GNN, GH-SolvGNN, GNNCat and SolvGNN-Cat models	192
A.14	Errors in the Vol. XIV of DECHEMA Chemistry Data Series	193
A.15	Derivation of the extended Margules equation	194
	Afterword	197

Introduction

1

” *We compete against problems, not against people.*

— **John B. Goodenough**
(Nobel Prize in Chemistry, 2019)

Humankind has a long history dealing with mixtures. Perhaps, the first examples of human-made separations can be traced back to the extraction of edible parts from plants and animals, a fundamental activity for the early humans' survival. The extraction and purification of metals from ores, an important step in the evolution of technology and society, and the distillation of alcohol from fermented substances also highlight the significant impact of mixture separation on human history [145]. Not only the task of separating mixtures has been important, also the combination of substances to create new mixtures has advanced society in different ways. The creation of new metal alloys with desired properties and the creation of novel mixtures for the development of plastics, paints and adhesives has shaped many technological advancements that have resulted in our today's standard of living.

Along with the technological and societal advancements that have taken us this far, new global challenges have arisen. Our society is consuming natural resources at a rate that exceeds our planet's capacity for regeneration, and this consumption is accelerating over time [44]. This results in a clear sustainability problem. Therefore, there is the need of shifting the current ways of production, distribution and consumption of resources and goods to meet the natural boundaries of our planet. Chemical processes, as the central means of today's society production system, are at the core of the sustainability goal. Industrial separation processes, specifically, account for 10-15% of the world's total energy consumption [147]. And, thermal separation technologies (i.e., distillation, evaporation and drying) alone account for approximately 80% of all energy consumed by industrial separations [93]. Our relationship of us, humans, with mixtures appears to be strong and important towards our sustainability objective. Just as we skillfully combine elements to create mixtures that advance society, we must also master the "art" of efficient separation to hopefully revert the environmental damage and keep pushing society forward in a sustainable way.

Initially, separation processes were simplistic, often relying on basic mechanical methods such as filtration, gravity separation, and manual sorting. These processes were fundamental in industries like mining, agriculture, and early chemical manufacturing. However, their design was mainly driven by empirical thinking that follows from common experience. With the rise of the Industrial Revolution on the 18th and 19th centuries, there was a need for more efficient and large-scale separation methods. This led to the development of more sophisticated separation equipment which could handle larger volumes and more complex mixtures. It was precisely around this time that chemical engineering was established as a field for dealing with the “industrial chemistry” [118]. The design of separation equipment such as distillation columns and evaporators started to take place in a more studied manner, but still relying mainly on empirical knowledge with very few instances of systematic knowledge in the form of physical property tables and constructor specifications [36]. Despite this, the sophistication of the separation equipment of that time was notable, and to some extent not radically different from the equipment used nowadays. In the 20th century, separation processes became fundamental in petrochemical, pharmaceutical, and food industries. More systematic approaches for process design started to arise such as the McCabe-Thiele method [104] for distillation column design. After World War II, there was an increased focus on optimizing processes for energy efficiency and capacity, driven by the economic expansion and growing environmental concerns. This era witnessed a stronger use of applied mathematics and the introduction of computer-aided design and simulation, allowing for more precise and efficient separation process design [118]. The 21st century has seen more integrated approaches across multiple spacial scales in which the process design paradigm is taken from the traditional *unit operation level* to the *phase-level* and even to the *molecular-level* on what is known as Computer-Aided Molecular and Process Design (CAMPD) [49]. The evolution of separation processes design is depicted in Fig. 1.1.

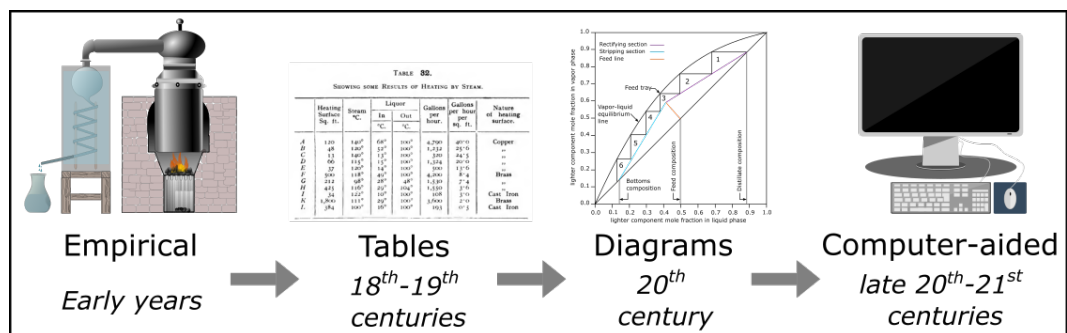


Fig. 1.1.: Depiction of the main techniques used for separation processes design throughout history.

1.1 Motivation

Separation processes of fluid mixtures take advantage of the differences on the thermodynamic and transport properties of the constituents to carry out the separation. These differences in the macroscopic (i.e., *phase-level*) properties are in turn a result of the constituents' differences at the *molecular-level*. Therefore, a relationship $f_{pure} : \mathcal{A} \rightarrow \mathcal{B}$ must exist such that, given the set of *molecular-level* characteristics \mathcal{A} (e.g., molecular weight, polarizability, molecular shape, configuration, atomic constitution...) of a chemical species, its *phase-level* properties \mathcal{B} (e.g., vapor pressure, melting point...) can be accurately predicted. In the paradigm of computer-aided process design, the impact of mathematically modeling f_{pure} is apparent as this would allow the estimation of relevant physical properties for chemical process calculations. Not only this would be highly valuable for the chemical engineer designing processes, but also for the chemist developing new products with desirable physical characteristics, and for the physical chemist who is trying to unravel the actual laws of nature. This is the reason why a considerable amount of research [22, 120, 83, 133, 8] has been dedicated to the modeling of f_{pure} since the pioneering efforts of Langmuir [94] by introducing the idea of group-contribution methods. This continues to have a key role within the Computer-Aided Molecular Design (CAMD) and CAMPD paradigms.

The accurate modeling of f_{pure} (despite being already challenging by itself) is only part of the overall challenge of constructing the mathematical foundations needed for the development of more sustainable separation processes. The reason is that f_{pure} only describes the information of pure chemical species. One would also need to accurately model the *phase-level* properties of the mixture itself. Similarly to the pure-component *phase-level* properties \mathcal{B} , a relationship $f_{mix} : \mathcal{C} \rightarrow \mathcal{D}$ must exist that maps the *molecular-level* characteristics of the mixture constituents together with its relative composition and physical state (i.e., pressure, volume and temperature) to its *phase-level* properties. While the physical P - V - T state of a fluid mixture and its composition can be characterized relatively well with the available measurement devices, it is the lack of understanding of the complex inter- and intramolecular interactions which prevents us from a smooth transition from the corresponding f_{pure} relationships of the mixture components to the actual f_{mix} relationship. The statement Abrams and Prausnitz wrote almost three decades ago remains true today - "*Despite attention for over a century from some of the best scientific minds, the goal of predicting mixture properties from pure-component properties alone remains elusive*" [3].

Ideally, the mathematical models for f_{pure} and f_{mix} need to be predictive in nature. The reason for this is that the chemical space that we face in designing novel processes using novel materials is enormous [129]. It is possible that a yet unexplored molecule or material within such vast chemical space could surpass its current industrial counterpart in terms of sustainability, cost-effectiveness, and efficiency. Exploring the entirety of the chemical space through experimental means is unfeasible. The situation is even worse when considering the realm of mixtures. The combinatorial nature of them causes the already extensive chemical space to expand to staggeringly immense proportions, rendering comprehensive experimental exploration even more impossible. As a matter of fact, if we assume that only 1,000 compounds are of interest, even for the most popular physical properties of mixtures like vapor-liquid equilibria, only around 1.2% of the experimental data is available [56]. Yet, the importance of mixtures is unarguable. It may not be a single "golden" molecule or material that propels the advancement of sustainable separation processes, but rather, the key could lie in a cleverly designed mixture.

To advance the design of sustainable separation processes, a deeper understanding of both *molecular-level* and *phase-level* physical phenomena of pure components and mixtures is crucial. However, what is particularly vital is expanding our knowledge about the interconnection between these two spatial scales. This "bridge" is key to effectively linking microscopic interactions with macroscopic behavior, thereby enhancing our ability to describe phase equilibria better and in turn to design more efficient and sustainable separation processes. A promising approach towards this goal is to utilize the available experimental data to offset the scarcity of system-specific information [103]. By combining the data with the current mechanistic understanding, hybrid mechanistic and data-driven models could potentially take us closer to the development of more sustainable separation processes.

In recent years, the field of data-driven modeling has witnessed remarkable progress, largely fueled by advancements in machine learning and deep learning. This surge in innovation can be attributed not only to the algorithms and techniques emerging from these domains, but also to the substantial improvements in computational capabilities [146], and to the trend towards open-sourcing technological frameworks [116, 1]. This latter development has notably lowered the entry barriers to the field, making these powerful tools more accessible and fostering a broader participation in this area of research and application. The question is whether this progress in data-driven modeling can also be transferred to the area of applied thermodynamics in combination with the available mechanistic understanding to support the general problem of molecular and mixture property prediction, specifically regarding the prediction of phase equilibria in separation processes.

1.2 Contributions of this thesis

Scope: The challenge of investigating the use of hybrid models for assisting separation process design is indeed broad. For the scope of this dissertation, the use of data-driven tools has mainly focused on investigating how graph neural networks, an arguably advanced machine learning technique, can assist the process of constructing foundational predictive models for fluid phase equilibria. The use of graph neural networks in this research is motivated by their recent success in predicting molecular properties, particularly in the field of drug discovery [163]. The role of accurately predicting phase equilibria of fluid mixtures within the context of separation processes design is remarkable [23]. Even a small improvement in the prediction accuracy can significantly impact the outcomes of the design. If a model falls short, there exist a risk of incorrectly predicting that separation will not occur when it actually will, or conversely, suggesting a viable separation for a system that does not exhibit such behavior.

In this study, the focus is restricted to systems operating under low to medium pressure levels. This limitation is due to the reliance on the concept of activity coefficients in the phase equilibria calculations conducted here. Therefore, systems at high pressure levels (for which it is well known that activity coefficient-based models do not work well [56]) are left out of the present scope. This dissertation also aims for the advancement of the understanding of different hybridization strategies that can be applied to combine the worlds of mechanistic modeling and data-driven modeling. This is particularly relevant in the context of applied thermodynamics, where the thermodynamic principles and constraints are fundamental and must be adhered to by any modeling framework.

Therefore, the contributions of this dissertation can be summarized by addressing the following set of questions:

- **Application of graph neural networks:** How can graph neural networks be leveraged to aid in the design of separation processes, particularly through the prediction of activity coefficients for phase equilibria calculations?
- **Effectiveness of hybrid models:** Do hybrid models, which combine mechanistic and machine learning approaches, offer more robust solutions for predicting phase equilibria of fluid mixtures compared to standalone models?
- **Exploration of hybrid model arrangements:** What configurations of hybrid models are most effective, and what factors contribute to their superiority in the context of phase equilibria prediction?

- **Integration into existing design frameworks:** Is it feasible to integrate these novel hybrid models into the existing methodologies used for separation process design, and if so, how can this be effectively achieved?

1.3 Thesis structure

The remainder of this dissertation is structured as follows:

Chapter 2. In this Chapter, the thermodynamic fundamentals of phase equilibria in fluid systems are presented along with the main approaches for modeling it. The role of molar excess Gibbs energy (g^E) models is described. Emphasis has been set on revising the concept of activity coefficients γ . The significance of activity coefficients in the infinite dilution region has been also explained. Additionally, this Chapter provides an overview of the concept of hybrid modeling, focusing on its two primary architectures: parallel and serial arrangements. Lastly, this Chapter delves into the fundamentals of graph neural networks, emphasizing the mechanisms that enable their training for the purpose of performing graph-level predictions.

Chapter 3. In this Chapter, an extensive comparison is presented between graph neural network models and the most widely-used predictive models for infinite dilution activity coefficients, specifically conducted at a temperature of 298.15 K. Additionally, the Chapter introduces the concept of hybrid parallel graph neural networks, examining their performance relative to their individual submodels. A number of additional isothermal studies are also provided.

Chapter 4. This Chapter generalizes the graph neural network models introduced in Chapter 3 for the prediction of activity coefficients at infinite dilution, by now allowing temperature variation. The temperature dependency is included by introducing the concept of hybrid serial graph neural networks. Here, mechanistic knowledge derived from the Gibbs-Helmholtz thermodynamic relation is used together with a series of graph neural networks for predicting the activity coefficients at infinite dilution. The extension of such model, referred to as the Gibbs-Helmholtz Graph Neural Network (GH-GNN), to more complex mixtures including ionic liquids and polymers is also presented.

Chapter 5. This Chapter progresses with the generalization of the graph neural network models to predict activity coefficients across varying temperatures and compositions. The concept of the hybrid serial arrangement is developed further by incorporating composition dependency through the extended Margules equation. The Chapter also presents a comparative analysis of this approach, which relies solely on infinite dilution data, against the commonly used UNIFAC-Dortmund model, developed using extensive and diverse types of experimental data. This comparison highlights their respective performances and limitations, particularly in the contexts of predicting vapor-liquid equilibria.

Chapter 6. This Chapter applies the previously introduced graph neural network-based models to practical scenarios in the design of separation processes. It specifically explores their application in the pre-selection of solvents for extractive distillation and liquid-liquid extraction. Additionally, the Chapter demonstrates the value of these type of models in supporting sustainable separation process design, particularly through their contribution to the design of a lignin process.

Chapter 7. In this Chapter, a summary of the key contributions of this thesis is presented, alongside an outlook that aims to steer future research and initiatives towards leveraging hybrid graph neural networks. This guidance is particularly focused on enhancing the development of more sustainable separation processes, highlighting the potential impact and applications of these advanced computational tools in the field.

” *Science advances one funeral at a time.*

— **Adapted from Max Planck**
(Nobel Prize in Physics, 1918)

This Chapter provides a comprehensive review of the foundational concepts that underpin the research presented in this work. The core of this work rests on three pillars: *phase equilibria* (with a special focus on the concept of activity coefficients) reviewed in Section 2.1, *hybrid semi-parametric modeling* reviewed in Section 2.2, and *graph neural networks* reviewed in Section 2.3. The subject of phase equilibria has long been a cornerstone in the field of chemical engineering due to its critical relevance in the typical tasks of a chemical engineer (e.g., process design, control and optimization). On the other hand, hybrid semi-parametric modeling, though less prevalent within the chemical engineering community, has seen a significant increase in interest over the past decade [76, 103, 77]. This surge is primarily attributed to recent advancements in data-driven modeling and computational techniques, coupled with the immutable physical constraints inherent in chemical engineering systems. Furthermore, the incorporation of graph neural networks into chemical engineering is a relatively recent development, marking a new frontier in the discipline. This Chapter aims to equip the reader with the necessary conceptual, thermodynamic, and mathematical background in these areas. However, for a detailed understanding of the overarching concepts on any of these areas, the reader may find it beneficial to consult the references provided herein.

2.1 Phase equilibria

When considering separation processes, separations by means of mechanical operations can be included. Examples of these include the separation of metals from a solid-waste stream by means of a magnetic field or the centrifugation of cream from milk. However, separation operations of fluid systems in the chemical industry are typically driven by an increase of the mass transfer rate of specific components

within a certain phase [145]. The speed at which separation occurs is determined by the rate of mass transfer, whereas the degree of separation itself is controlled by thermodynamic equilibrium.

Thermodynamic equilibrium of a multi-component mixture between Π phases is defined by the fulfillment of the following conditions

$$T^{(1)} = T^{(2)} = \dots = T^{(\Pi)} \quad (2.1)$$

$$P^{(1)} = P^{(2)} = \dots = P^{(\Pi)} \quad (2.2)$$

$$\mu_i^{(1)} = \mu_i^{(2)} = \dots = \mu_i^{(\Pi)} \quad \forall i \in \mathcal{CO} \quad (2.3)$$

which are referred to as the thermal (Eq. 2.1), mechanical (Eq. 2.2) and material (Eq. 2.3) equilibrium conditions. Here, \mathcal{CO} refers to the set of N mixture components. The conditions of equal temperatures (Eq. 2.1) and pressures (Eq. 2.2) are rather intuitive because of their formulation in terms of variables that seem closer to our daily life experience. By contrast, the condition of material equilibrium is formulated in terms of the chemical potential μ of each species $i \in \mathcal{CO}$ in the mixture for all the Π phases. The concept of chemical potential is much less intuitive compared to the concepts of temperature and pressure, and it is defined as the rate of change in the system's energy due to a change in the composition of a specific component. It is important to note that while the concepts discussed herein are relevant to the equilibrium of solid, liquid, and vapor phases, the focus here predominantly lies on the fluid phases (liquid and vapor). Unlike the solid phase, which maintains its shape under shear stress, fluid phases are characterized by their ability to deform and flow when subjected to such stress [56].

2.1.1 Thermodynamic fundamentals

Mixture properties

When addressing mixtures of N components, it is essential to consider the influence of composition on the thermophysical properties of the mixture, in addition to the effects of temperature and pressure. Historically, the modeling of mixture properties

has been approached through three primary mechanisms [56] that are explained briefly in the following:

1) Modeling the property change upon mixing: The idea here is to model the difference between the property of the real mixture and the weighted average of the pure component properties according to their molar fraction in the mixture. Therefore, for any molar property m we can write its property change upon mixing as

$$\Delta m = m - \sum_i^N z_i m_i \quad (2.4)$$

where z_i stands for the mole fraction of component i in the mixture, and m_i refers to the molar property of the pure component i .

2) Introducing the partial molar property: This approach relies on understanding how individual components contribute to the overall mixture property of interest. The idea is to model how a single component, infinitesimally added to a mixture while maintaining everything else constant, affects the overall system's property M . The definition of a partial molar property is then given by

$$\bar{m}_i \equiv \left(\frac{\partial M}{\partial n_i} \right)_{T,P,\mathbf{n}_{\neq i}} \quad (2.5)$$

where $\mathbf{n}_{\neq i}$ refers to the vector of number of moles for all mixture components except component i .

By having access to the partial molar property of each component in the mixture, one can calculate (as shown by the Euler theorem for homogeneous functions, Appendix A.1) that the mixture property is equal to the weighted average of the partial molar properties according to their mole fraction (Eq. 2.6).

$$m = \sum_i^N z_i \bar{m}_i \quad (2.6)$$

3) Introducing the excess property: This method involves the use of excess properties, which are defined as the difference between the actual property of the mixture and the property predicted by an ideal model (Eq. 2.7). Excess properties help in quantifying the deviations from ideality, offering a measure of the different interactions between different components in the mixture.

$$m^E = m - m^{id} \quad (2.7)$$

Chemical potential and fugacity

If we consider a closed system consisting of only one phase, its change in Gibbs energy G in differential form is given by

$$dG = dH - TdS - SdT = VdP - SdT \quad (2.8)$$

where H , S , T , V and P stand for the enthalpy, entropy, temperature, volume and pressure, respectively. If now material variation is allowed into the system (i.e., an open system), the amount of each one of the components in the system needs to be included as additional variables of the thermodynamic potential. Thus, the Gibbs energy in an open system is a function of temperature, pressure and composition $G = f(T, P, n_1, n_2, \dots, n_N)$, and the total differential can be written as

$$dG = \left(\frac{\partial G}{\partial T}\right)_{P, \mathbf{n}} dT + \left(\frac{\partial G}{\partial P}\right)_{T, \mathbf{n}} dP + \sum_i^N \left(\frac{\partial G}{\partial n_i}\right)_{T, P, \mathbf{n}_{\neq i}} dn_i \quad (2.9)$$

where $\mathbf{n} = [n_1, n_2, \dots, n_N]$ is the vector of number of mols for all components in the mixture, and $\mathbf{n}_{\neq i}$ stands for the vector of number of mols for all components except component i . By comparison of Eq. 2.9 to Eq. 2.8, it is clear that

$$\left(\frac{\partial G}{\partial P}\right)_{T, \mathbf{n}} = V \quad (2.10)$$

$$\left(\frac{\partial G}{\partial T}\right)_{P, \mathbf{n}} = -S \quad (2.11)$$

and thus, we can introduce the concept of chemical potential as being the last partial derivative of the right hand side of Eq. 2.9

$$\mu_i = \left(\frac{\partial G}{\partial n_i} \right)_{T, P, \mathbf{n}_{\neq i}} = \bar{g}_i \quad (2.12)$$

which corresponds to the partial molar Gibbs energy \bar{g}_i . Therefore, the chemical potential of a substance i in a mixture can be described as the rate of change in the system's Gibbs energy when one molecule of i is added to the system at constant T and P , holding the amount of the rest of components also constant.

At the same time, we can express the chemical potential in terms of an auxiliary variable called fugacity f_i (derivation in Appendix A.2) as

$$\mu_i(T, P, \mathbf{z}) = \mu_i^{0, pure}(T, P^0) + RT \frac{f_i(T, P, \mathbf{z})}{f_i^{0, pure}(T, P^0)} \quad (2.13)$$

where the superscript 0 refers to the reference state at an arbitrary pressure P^0 , the superscript *pure* indicates the properties of the pure compound i , \mathbf{z} is the vector of molar ratios for all mixture components, and R is the universal gas constant. The auxiliary variable fugacity needs to meet the limiting condition of the ideal gas

$$\lim_{P \rightarrow 0} f_i(T, P, \mathbf{z}) = z_i P \quad (2.14)$$

Since, for a multi-phase system in equilibrium, the temperature and pressure are the same in all phases, the material equilibrium condition (Eq. 2.3) can be expressed in terms of the fugacities f using Eq. 2.13 as

$$f_i^{(1)} = f_i^{(2)} = \dots = f_i^{(\Pi)} \quad \forall i \in \mathcal{CO} \quad (2.15)$$

The introduction of the chemical potential as a result of studying (chemical) systems with material exchange in the framework of classical thermodynamics was indeed an impactful contribution of Gibbs [51].

2.1.2 Activity coefficients

For simplification, another auxiliary variable is defined as the ratio of the fugacities that appear in Eq. 2.13 (or, in general terms, as the ratio of the fugacities with an arbitrary reference state). This variable is called the activity a_i

$$a_i \equiv \frac{f_i}{f_i^0} \quad (2.16)$$

If the reference state is the ideal gas at the same temperature, pressure and composition, the activity corresponds to the fugacity coefficient

$$\phi_i(T, P, \mathbf{z}) = \frac{f_i(T, P, \mathbf{z})}{z_i P} \quad (2.17)$$

Similarly, if the reference point is the ideal solution, the activity corresponds to the activity coefficient

$$\gamma_i(T, P, \mathbf{x}) = \frac{f_i(T, P, \mathbf{x})}{x_i f_i^{pure}(T, P)} \quad (2.18)$$

where x_i refers to the molar fraction of component i in the (liquid) solution. From Eq. 2.18 the limiting condition of $\gamma = 1$ when $x_i = 1$ is clear. The ideal solution is such that the fugacity of component i in the mixture f_i is equal to the fugacity of the pure component i weighted by the molar fraction (i.e., $f_i = x_i f_i^{pure}$). By combining Eq. 2.18 and Eq. 2.16 the relationship between activity and activity coefficient can be found

$$a_i = x_i \gamma_i \quad (2.19)$$

Intuitively, and by observing Eq. 2.18, the activity coefficient can be regarded as a measure of the deviation from the ideal behavior. This deviation occurs when the assumptions of the ideal solution are no longer valid. In particular, the ideal solution model assumes that all interactions between molecules are the same and, as a result, the partial pressures are equal to the molar proportion of the saturation pressure of the corresponding pure compound P_i^{sat} . This is exactly Raoult's law

$$P_i = x_i P_i^{sat} \quad (2.20)$$

where x_i stands for the molar fraction of component i in the liquid phase.

Therefore, in order to solve the material equilibrium condition (Eq. 2.15) there exist two approaches corresponding to how the fugacities are calculated using Eq. 2.17 and Eq. 2.18. These approaches are known as the " ϕ " and " γ " approaches, respectively. The latter approach is specially useful in the context of liquid solutions,

either within vapor-liquid or liquid-liquid equilibria calculations. For vapor phases the former approach is used. Therefore, for a vapor-liquid equilibrium calculation we would have that

$$\gamma_i x_i f_i^{pure}(T, P) = \phi_i^V y_i P \quad (2.21)$$

where the superscript V refers to the vapor phase, and, with the assumption that the molar liquid volume v_i^L is incompressible (this is a good approximation far from the critical point), the reference state fugacity is given by

$$f_i^{pure}(T, P) = \phi_i^{sat} P_i^{sat} \exp \frac{v_i^L (P - P_i^{sat})}{RT} \quad (2.22)$$

where the superscript L refers to the liquid phase.

At conditions where the difference between the system's pressure P and component's saturation pressure P_i^{sat} is not very large, the exponential term, known as the Poynting factor, is close to 1. Moreover, for non-associating compounds, the fugacity coefficient at saturation conditions ϕ_i^{sat} and the fugacity coefficient of the same compound in the vapor phase ϕ_i^V are very similar. Therefore, for many practical applications vapor-liquid equilibrium (VLE) can be approximated by

$$\gamma_i x_i P_i^{sat} \approx y_i P \quad (2.23)$$

Similarly, liquid-liquid equilibrium (LLE) between phase (1) and phase (2) requires the fulfillment of the isoactivity condition given by

$$\gamma_i^{(1)} x_i^{(1)} = \gamma_i^{(2)} x_i^{(2)} \quad (2.24)$$

Therefore, it becomes clear that the role of the activity coefficient for fluid phase equilibria calculations is central. The activity coefficient depends mainly on the temperature of the mixture T and its composition \mathbf{x} , and, to a lesser extent, on the pressure according to the following expressions

$$\left. \frac{\partial \ln \gamma_i}{\partial (1/T)} \right|_{P, \mathbf{x}} = \frac{\bar{h}_i^E}{R} \quad (2.25)$$

$$\left. \frac{\partial \ln \gamma_i}{\partial P} \right|_{T, \mathbf{x}} = \frac{\bar{v}_i^E}{RT} \quad (2.26)$$

$$\sum_i^N x_i d \ln \gamma_i = 0 \quad (2.27)$$

The derivation of these expressions from the excess properties is available in Appendix A.3. It is also important to mention that Eq. 2.27 is valid only for the isobaric and isothermal case. Moreover, this expression is not (strictly speaking) valid for binary mixtures due to the Gibbs phase rule constraint. Despite this, for the binary case, it provides a fair approximation when applied to the isothermal case given the small dependency of γ_i on P [122].

There are two last expressions that are worth mentioning here, which relate the activity coefficient with the partial molar excess Gibbs energy (derivation in Appendix A.4)

$$\bar{g}_i^E = \frac{\partial G^E}{\partial n_i} = RT \ln \gamma_i \quad (2.28)$$

and the molar excess Gibbs energy

$$g^E = RT \sum_i^N x_i \ln \gamma_i \quad (2.29)$$

Since the assumptions of the ideal solution (i.e., intermolecular interactions within the mixture are identical, and all molecules are of the same size and shape) are met for pure compounds, the excess properties for pure substances are zero by definition. This defines the following boundary condition for the molar excess Gibbs energy

$$g_i^E \rightarrow 0 \quad \text{when} \quad x_i \rightarrow 1 \quad (2.30)$$

The condition given by Eq. 2.30 has to be met by any model that predicts g_i^E . In the course of history several models have been proposed for predicting g_i^E , from which one can calculate the needed activity coefficients for the posterior phase equilibria calculations. Among the most influential and widely used excess Gibbs energy models are the Wilson [164], NRTL [128], and UNIQUAC [3] models.

At conditions where the component i is infinite diluted, the activity coefficient is represented as γ_i^∞ , and it is referred to as the infinite dilution activity coefficient (IDAC). This coefficient, when reported for binary mixtures, is of special theoretical interest because it provides a measure of the interactions that provoke the non-ideality of substances in liquid mixtures. Moreover, its practical significance spans a wide range of chemical processes, especially for the ones achieving high product purities, where conditions approach infinite dilution. Moreover, γ_i^∞ values can be used for parameterizing models that predict the whole range of compositions [144, 139, 25]. Similarly, in extraction processes, the pre-selection of appropriate solvents often depends on the values of γ_i^∞ [25, 11]. Beyond process design, γ_i^∞ values are also integral to safety and environmental protection studies, where understanding the solute's behavior at infinite dilution can inform hazard assessments and mitigation strategies [140].

2.2 Hybrid data-driven and mechanistic modeling

To improve the design of separation processes for enhanced efficiency and sustainability, leveraging accurate and computationally efficient models for predicting phase equilibria is essential. While one approach involves constructing models based solely on our mechanistic understanding of the molecular processes driving phase equilibria, this method is time-intensive and demands a high level of expertise that is scarce [37]. Furthermore, our understanding of molecular processes and complex intermolecular interactions remains incomplete in several areas, such as emulsions and dispersions [37]. Consequently, an alternative strategy is to utilize the knowledge embedded in thermophysical data through data-driven models. The development of these models offer greater flexibility compared to purely mechanistic ones, and, hence, tend to be more accessible to a broader community. However, embedding the physical and thermodynamic constraints into data-driven models is challenging [76]. Therefore, hybrid models, which integrate fundamental principles with data-driven parts, represent a promising direction for model development of fluid phase equilibria and separation processes in general [103]. This type of models are increasingly recognized for their potential to significantly improve both the accuracy and efficiency of predictions, while retaining the constraints of the physical systems.

In fact, practically all chemical process models (at the various spacial scales) are composed in a hybrid manner. Usually, a mechanistic part in the form of algebraic or differential equations is coupled with parameters that are fitted to experimental data.

This is the case also in many thermodynamic models (e.g., NRTL [128], UNIQUAC [3]). Hence, such models might be regarded as hybrid models already. However, the emphasis here is for the so-called “hybrid *semi-parametric* models” which combine a *parametric* part (models with a defined set of parameters) with a *non-parametric* part (models that can adapt flexibly to the data and do not have a defined set of parameters) [54]. Hybrid semi-parametric models leverage the strengths of both approaches. They use parametric models to capture known relationships within the data based on theoretical or phenomenological evidence (e.g., conservation laws, kinetic behavior), while employing non-parametric (or more generally, data-driven) models to learn and adapt to complex data patterns that cannot be easily captured by parametric forms. This hybrid approach aims to balance the interpretability and simplicity of parametric models with the flexibility and adaptability of non-parametric models. In the context of separation processes, McBride et al. [103] provides an overview of several hybrid semi-parametric models that have supported the development of separation processes.

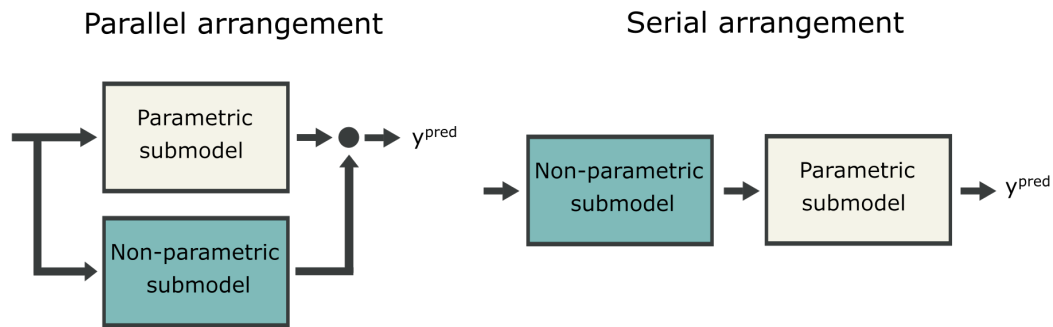


Fig. 2.1.: Main hybrid semi-parametric model arrangements.

Hybrid semi-parametric models can be structured in two primary configurations: parallel and serial. These configurations are illustrated in Fig. 2.1, with further details provided in the subsequent Subsections. Each arrangement offers a unique approach to integrating the strengths of both parametric and non-parametric models, enabling flexible and robust modeling of complex data relationships.

2.2.1 Parallel arrangement

In the parallel configuration of hybrid semi-parametric models, as illustrated on the left side of Fig. 2.1, the model incorporates both parametric and non-parametric submodels working in tandem. The parametric submodel is responsible for making the direct prediction based on a predefined mathematical framework, while the

non-parametric submodel enhances this prediction by applying a correction. This correction typically manifest as additive adjustments, although multiplicative coupling can also be employed to refine the predictions. The final prediction of the hybrid model is then the prediction of the parametric submodel corrected by the non-parametric counterpart. An advantage of this arrangement is that the training of the non-parametric part can be decoupled from the development of the parametric submodel. This means that, once the parametric model has been developed, the non-parametric model can be trained from the residuals of the parametric submodel in a subsequent step.

Interestingly, this parallel arrangement restates a fundamental principle often observed in thermodynamics, where the concept of applying “corrections” to theoretical models is commonplace. For instance, in thermodynamics, the compressibility factor is used to quantify deviations from ideal gas behavior, and the activity coefficient provides adjustments from the ideal solution model. Such corrections are vital for aligning theoretical models more closely with real-world observations, improving the accuracy of predictions and the understanding of complex phenomena. The synergy between parametric predictions and non-parametric corrections in hybrid semi-parametric models mirrors this thermodynamic approach, enabling the model to accommodate real-world complexities that might not be fully captured by traditional parametric models alone. This hybridization strategy can be motivated by a lack of mechanistic understanding or simply by computational convenience. This parallel arrangement not only enhances model flexibility and predictive power but also highlights the value of integrating diverse modeling strategies to address the complexity of certain physical phenomena.

2.2.2 Serial arrangement

The serial arrangement of hybrid semi-parametric models is designed by sequentially linking the two submodels, where the output from the first serves as the input for the second to generate the ultimate prediction. This arrangement allows for a layered approach to modeling, where the insights or predictions from one layer inform and refine the processing of the subsequent layer. Although it is feasible to structure this configuration with the parametric component leading into the non-parametric, it is more commonly arranged with the non-parametric part preceding the parametric one [54, 103], as illustrated on the right side of Fig. 2.1. This preferred sequence, where the initial, flexible non-parametric model’s predictions are further refined and contextualized by the parametric model, leverages the strengths of both approaches.

The rationale behind the serial arrangement is that the non-parametric model, due to its flexibility and fewer *a priori* assumptions, can capture complex patterns and relationships in the data that might not be readily apparent or easily modeled by parametric methods. These findings are then fed into the parametric model, which can incorporate this embedded information into a more structured, theory-driven framework. This sequential embedding allows the hybrid model to benefit from the non-parametric model's ability to detect and model intricate data complexities, while the subsequent parametric model applies theoretical principles to ensure that the predictions adhere to known behaviors or relationships.

2.3 Graph neural networks (GNNs)

Graph neural networks (GNNs) refer to a class of artificial neural networks designed specifically for processing data that is represented as graphs. This approach is particularly beneficial for handling structured data, where, besides the elements, the relationships between elements is also important. Unlike traditional artificial neural networks, which assume that data points are independent and identically distributed (i.i.d.), GNNs learn from the complexity of graph structures, making them ideal for a wide range of applications where data relationships play a crucial role [24].

A graph is usually represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} is the set of nodes in the graph and \mathcal{E} is the set of edges. Nodes (and edges) are usually attributed, which means that they have an associated vector of features. The goal of a GNN is to learn a useful graph representation (embedding) that allows us to perform desired downstream tasks (e.g., graph property prediction) or that allows us to reconstruct the original graph from the embedding space [167]. This graph representation should take the form of a dense and continuous vector, and should exist in an embedding space that conserve the relationships of the graph structure data as relative distances within the space. GNNs learn such graph representations through a series of operations (layers) where each operation (layer) updates the node embeddings by aggregating representations of its neighboring nodes. Therefore, we can write a GNN layer l as

$$f(\mathbf{a}_v^l) = \Xi \left(\mathbf{a}_v^l, \square_{w \in \mathcal{N}(v)} \psi(\mathbf{a}_v^l, \mathbf{a}_w^l) \right) \quad (2.31)$$

where f is the GNN operation, \mathbf{a}_v^l is the embedding of node v at layer l , $\mathcal{N}(v)$ is the set of indices of the neighbouring nodes of node v , \square is a permutation-invariant aggregation operator (e.g., sum, mean), ψ is a *message passing* function that takes

the embeddings of node v and node w to construct a message that is sent to update the features of node v , and Ξ is an *updating* function. Both ψ and Ξ are learnable functions, which means that they are parameterized differentiable functions. In Eq. 2.31, only the node embeddings take part in the operations. However, edge features and other graph information can also be included as part of the learning framework of a GNN.

Since their introduction [143], GNNs have evolved into several variants depending on the different mechanisms used for performing the three main steps of the general message passing framework: *message passing*, *aggregation* and *updating*. Perhaps, the most influential GNN architectures are the Graph Convolutional Network (GCN) [82], the Graph Attention Network (GAT) [156] and the Message Passing Neural Networks (MPNNs) [52].

Consequently, by layering multiple layers as depicted in Eq. 2.31, a GNN can encapsulate both the feature information and the structural topology of graphs through supervised learning. Specifically, in tasks aimed at predicting properties of entire graphs, the GNN is trained on a data set comprising graphs along with their respective labels. The training process involves minimizing a loss function (e.g., mean squared error), thereby allowing the model to adjust the parameters of the ψ and Ξ functions through back-propagation and the application of a gradient descent-based optimization algorithm. This learning framework is known as supervised learning [19], and enables the GNN to effectively learn and leverage the complex patterns present within both the features and the topology of graphs, facilitating accurate predictions of graph properties.

To facilitate graph-level predictions (either regression or classification), a global pooling operation is necessary. This operation aggregates the representations of all nodes (and possibly edges and global information) in the graph, commonly derived from the final layer L of a GNN, into a unified vector representation of the graph. The aggregated graph-level representation, \mathbf{a}_g , enables the application of subsequent prediction tasks on the graph as a whole. The global pooling mechanism can be mathematically expressed as

$$\mathbf{a}_g = \mathcal{P}(\{\mathbf{a}_v^L | v \in \mathcal{V}\}) \quad (2.32)$$

where \mathcal{P} denotes the global pooling function, which operates over the set of node representations \mathbf{a}_v^L from the last layer L of the GNN, for all nodes v within the graph \mathcal{V} . This function also needs to be permutation invariant since the ordering of the nodes in a graph is not defined.

Predicting Isothermal Infinite Dilution Activity Coefficients

” *The physics of molecular condensed phases are not now and may never be known sufficiently well to allow attainment [of the precise prediction of physical properties from molecular structure alone] by tractable theoretical formulations.*

— **Arnold Bondi**

(Scientist for Shell Development Co.)

As discussed in Chapter 2, understanding the role of activity coefficients is crucial for accurately modeling fluid phase equilibria. Specifically, infinite dilution activity coefficients (IDACs) are particularly important. They provide insights into how different molecules interact with one another, which is essential both for theoretical understanding and practical applications, such as selecting the best solvents for extractive processes. As also discussed in Section 2.1.2, to model activity coefficients effectively, one must consider mainly their dependence on temperature and composition. Starting with the modeling of isothermal IDACs presents a convenient starting point given that the effects of composition and temperature are held constant, allowing the focus to be solely on the interactions between different mixture components. The analysis presented in this Chapter centers mainly on the standard temperature 298.15 K, and concludes with a brief extension to other isothermal cases.

There exist different models for predicting IDACs, which can broadly be categorized into phenomenological and machine learning-based models. Historically, phenomenological models have been the go-to choice, benefiting from decades of refinement, application, and validation. This preference stems from their long-standing development and proven track records, in contrast to the relatively newer machine learning techniques. Despite their established use, phenomenological models are not without flaws. This issue is thoroughly discussed in the study by Brouwer et al. [28], highlighting the limitations inherent in phenomenological models. Meanwhile, the growth in computational capabilities and the expansion

of comprehensive databases have paved the way for machine learning methods to emerge as a promising alternative.

Phenomenological models: They can be organized into four main types: solvation models, group contribution methods, linear solvation energy relationships (LSER), and COSMO-RS.

- *Solvation models* account for the solute-solvent interactions based on specific model parameters that depend on the specific molecules at hand. The more parameters they include, the better the predictions tend to be. The most popular models include the Hildebrand model [71], the Hansen solubility model [65] and the Modified Separation of Cohesive Energy Density (MOSCED) model [153]. The mathematical formulation of these specific models can be found in Appendix A.5.
- *Group contribution models* estimate activity coefficients based on the molecular structure by using the assumption that groups in the molecule contribute independently (and via addition) to non-ideality despite of their surrounding environment. The most popular model under this category is the original UNIFAC model [47] and the modifications thereof, UNIFAC-Lyngby [95, 81] and UNIFAC-Dortmund [161, 57, 58, 59]. The mathematical formulation of them can be found in Appendix A.6.
- *Linear Solvation Energy Relationship models* correlate a set of solute and solvent descriptors to the partition coefficients of the solute into different phases. This partition coefficients can be then used to estimate IDACs. Perhaps, the most popular model of this kind is the Abraham model [2] which is described in Appendix A.7.
- Lastly, *COSMO-RS* [42, 124], a model grounded in quantum chemistry, utilizes the conductor-like screening model (COSMO) theory to estimate molecular interactions. This approach enables the prediction of fluid phase equilibria by accounting for a range of interaction types, including electrostatic forces, hydrogen-bonding, and van der Waals interactions.

Machine learning models: There are different data-driven models that have been proposed in the literature for the prediction of IDACs. Specially, in the last years during the development of this thesis, the interest in investigating this type of methods have increased significantly. We can categorize machine-learning based IDAC models into three main groups as follows:

- Numerous *quantitative structure-property relationship (QSPR) models* for predicting IDACs have been proposed in the literature [106, 53, 43, 4, 113, 13] for a relatively long time. Developing these type of models predominantly requires two foundational steps: selecting appropriate molecular descriptors and employing a regression technique. While for the latter there are many available options (e.g., linear regression, Gaussian process, support vector machine), the former step requires extensive exploration of descriptors and specific expert knowledge to guide the selection process.
- *Matrix completion methods (MCM)* and *Tensor Completion Methods (TCM)* have recently emerged as a promising data-driven alternative in the prediction of IDACs [74, 75, 33, 34, 32, 151] and other thermophysical [67] and transport [60, 61] properties. This type of models formulate predictions using a partially completed solute-solvent matrix (i.e., solutes as columns and solvents as rows) of IDAC values. The application scope for MCMs, however, is restricted by the specific solute-solvent matrix it employs, and the precision of its predictions is deeply influenced by the size of the data set and the inter-data correlations [33].
- *Deep-learning models* have also recently gained interest in the prediction of IDACs. This type of models aim to overcome the limitation of manually selecting molecular descriptors by providing an end-to-end framework that can implicitly learn appropriate embeddings (descriptors) using a supervised learning scheme. The two main deep-learning approaches for the prediction of IDACs are the methods based on graph neural networks and transformers. The former includes the pioneering work developed in the course of this thesis [138] along with an independent work by Felton et al. [45], and further developments [125, 137, 136, 132]. Methods leveraging transformers, as primarily developed by Winter et al. [166, 165], incorporate a mechanism called attention, allowing the model to weigh the importance of different parts of the input data differently.

3.1 Data sources and data cleaning

The data set utilized for the work presented in the following Sections is compiled from existing literature sources, originating from the compilation by Brouwer et al.[28]. The original compilation was corrected for containing a few data point errors on April 2023 [27]. This collection encompasses binary IDACs determined

experimentally at a temperature of 298.15 K and for systems including organic molecules and ionic liquids. For the work presented in this Chapter, all data points involving ionic liquids were removed. The experimental measurement techniques used for the determination of the IDACs varied, incorporating both non-analytical approaches (e.g., ebulliometry, and dew-point temperature measurement) and analytical strategies (e.g., gas-liquid chromatography retention measurements). Analytical techniques generally offer more reliable confidence intervals than non-analytical methods. Nevertheless, due to the limited data availability and the prevalent omission of confidence intervals in the original experimental works, this study considers data gathered through both methodologies.

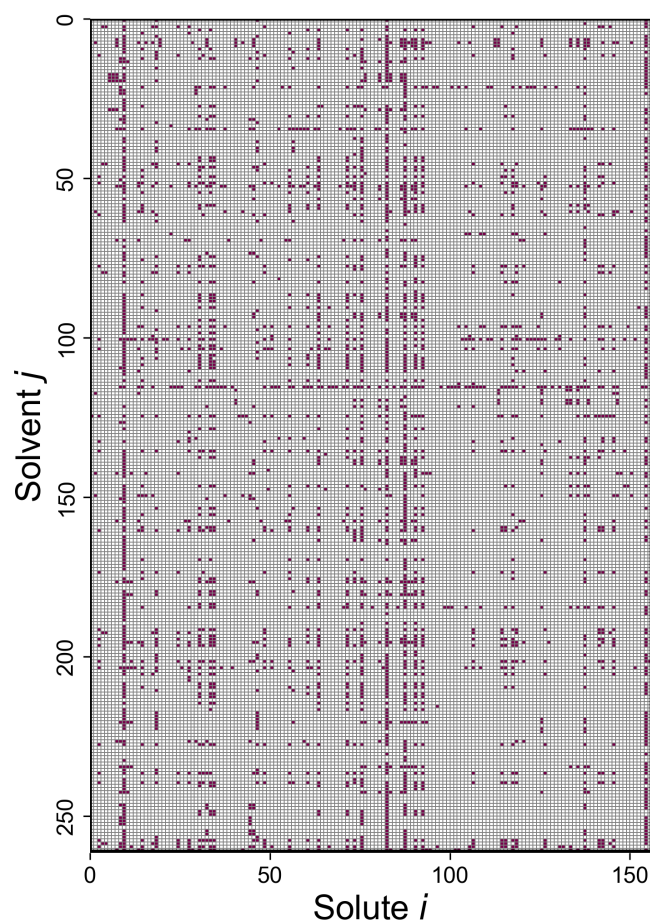


Fig. 3.1.: Solute-solvent matrix representation of the available IDAC data at 298.15 K on the *Brouwer data set*. Available IDAC data is represented by colored entries.

After removing all ionic liquid data, the database encompasses a collection of 4,460 data points, representing combinations from 156 solutes and 261 solvents, spanning across binary mixtures. Despite the extensive data set, there exists redundancy within the entries, with multiple data points corresponding to identical solute-

solvent systems. After identifying these overlaps, the unique system count is refined to 2,810 distinct pairs, which accounts for approximately 6.9% of the total binary combinations, as exemplified in Fig. 3.1. The arithmetic mean was computed for duplicated entries to achieve a singular, representative IDAC value for each solute-solvent system. The cleaned data set is referred to as *Brouwer data set*.

For the purposes of scaling and facilitating more intuitive analysis, the natural logarithm of the IDACs, denoted as $\ln(\gamma_i^\infty)$, was preferred over the actual γ_i^∞ values. This transformation is particularly advantageous as these scaled coefficients align more closely with the calculation of chemical potentials (cf. Eq. 2.28), and effectively scale the wide range of possible IDAC values.

3.2 Molecular graphs

Both solutes and solvents are represented as graphs, which are specifically termed *molecular graphs* in this domain. Initially, solutes and solvents were encoded using the SMILES notation [162] that represent molecular structures as strings. The cheminformatics toolkit `RDKit` [127], version 2021.03.1, was utilized to extract atom and bond characteristics for each molecule. Then, a graph object was constructed using `PyTorch Geometric` [46], where nodes and edges represent atoms and covalent bonds, respectively. In order to have more condensed graph representations, only non-hydrogen atoms are explicitly included as nodes in the graph. Each node and edge in this molecular graph is attributed with a vector detailing the previously mentioned atom and bond characteristics obtained from `RDKit`. The selection of features for nodes and edges, as outlined in Tables 3.1 and 3.2, respectively, was guided by the objective to capture distinct atomic and bond features within molecules, and was inspired by methodologies previously described in the literature [52].

The dimensionality of the node feature vector $\mathbf{a} \in \{0, 1\}^{25}$ and edge feature vector $\mathbf{b} \in \{0, 1\}^6$ is given by the sum of the dimensions indicated in Tables 3.1 and 3.2, respectively. Specifically, binary features that categorize into one of two groups (e.g., “Ring” and “Aromatic” for node features as shown in Table 3.1, and “Conjugated” and “Ring” for edge features in Table 3.2) are encoded with a single binary value, indicating the presence (1) or absence (0) of each category. This binary encoding approach, known as one-hot encoding, was used for all features. In essence, one-hot encoding converts categorical variables into a binary vector, ensuring each category is represented by a unique entry where only one element is marked with a value of 1, and all others are set to 0. By treating categories as separate dimensions without

Tab. 3.1.: Atomic features defining the initial feature vector of nodes in the molecular graphs constructed from the Brouwer data set. The dimension of the corresponding one-hot encoded feature is also shown.

Feature	Description	Dimension
Atom type	[C, Br, Cl, N, O, I, S, F, P]	9
Ring	Is the atom in a ring?	1
Aromatic	Is the atom part of an aromatic system?	1
Hybridization	[sp, sp ² , sp ³]	3
Bonds	Number of bonds the atom is involved in [1,2,3,4]	4
Charge	Atom's formal charge [0,-1,1]	3
Attached Hs	Number of bonded hydrogen atoms [0,1,2,3]	4

Tab. 3.2.: Bond features defining the initial feature vector of edges in the molecular graphs constructed from the Brouwer data set. The dimension of the corresponding one-hot encoded feature is also shown.

Feature	Description	Dimension
Bond type	[Single, double, triple, aromatic]	4
Conjugated	Whether the bond is conjugated	1
Ring	Whether the bond is part of a ring	1

implying any inherent order, one-hot encoding facilitates the accurate and unbiased processing of non-numeric information, enhancing the performance of machine learning algorithms.

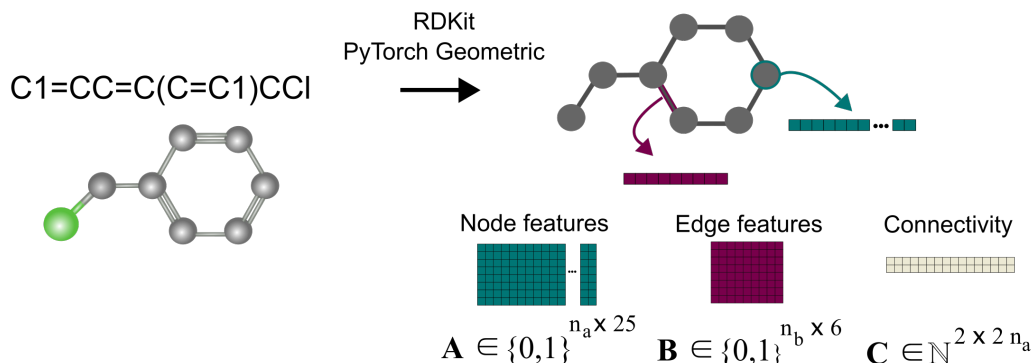


Fig. 3.2.: Schematic representation outlining the process of generating molecular graphs from SMILES strings.

A molecular graph is represented by the atom feature matrix $\mathbf{A} \in \{0,1\}^{n_a \times 25}$ and the bond feature matrix $\mathbf{B} \in \{0,1\}^{n_b \times 6}$, where n_a and n_b denote the number of nodes and edges, respectively. These matrices are formed by stacking the corresponding node and edge feature vectors. The connectivity of the molecular graph is represented by a connectivity matrix $\mathbf{C} \in \mathbb{N}^{2 \times 2 n_a}$, which contains the indices of source and receiver nodes within the graph. The first row of this matrix enumerates the indices of the

source nodes, while the second row specifies the indices for the target nodes. In the context of molecular graphs, directed edges (covalent bonds) are not physical. This aspect explains the dimensions of the matrix \mathbf{C} , where $2n_b$ reflects the bidirectional nature of covalent bonds, thereby permitting nodes to serve simultaneously as sources and receivers. The construction process of molecular graphs from SMILES strings is depicted in Fig. 3.2.

This approach to modeling molecules as graphs offers a distinct alternative to other molecular representation techniques, such as the identification of functional groups employed in group contribution methods. In contrast to the manual definition of groups, which relies heavily on the expertise of physical chemists, graph-based representations enable the application of techniques like graph neural networks (GNNs). GNNs have the capability to compute molecular representations, optimizing them based on the atomic and bond information data and their relevance for accurately predicting $\ln \gamma_i^\infty$. This end-to-end learning process facilitates a more data-driven approach to molecular representations and mixture-related property prediction compared to the mechanistic models described at the beginning of this Chapter.

3.3 Graph neural network

In an initial approximation, the molecular graphs representing both the solute and the solvent are independently processed by distinct GNNs. Specifically, one GNN is dedicated to generating a vectorial embedding for the solute, while a separate GNN is tasked with producing the vectorial embedding for the solvent. This approach ensures that the unique structural characteristics of each component are effectively captured and represented through their respective embeddings. Subsequently, the embeddings derived for both the solute and the solvent are concatenated to construct a vectorial representation of the isothermal binary-mixture at infinite dilution. This mixture embedding is utilized as input to a multi-layer perceptron (MLP) for regressing the corresponding $\ln \gamma_i^\infty$ value.

The same architecture was used for both GNN models, the one processing the solvent and the one processing the solute. First, the initial node embedding $\mathbf{a}_v^{(0)}$ of each node $v \in \mathcal{V}$ is transformed using a single-layer neural network $\phi_0 : \{0, 1\}^{25} \mapsto \mathbb{R}^{d^{(l)}}$ with the Leaky ReLU activation function to map the dimensions of the original node feature vector to the dimensions of the node embeddings during message-passing.

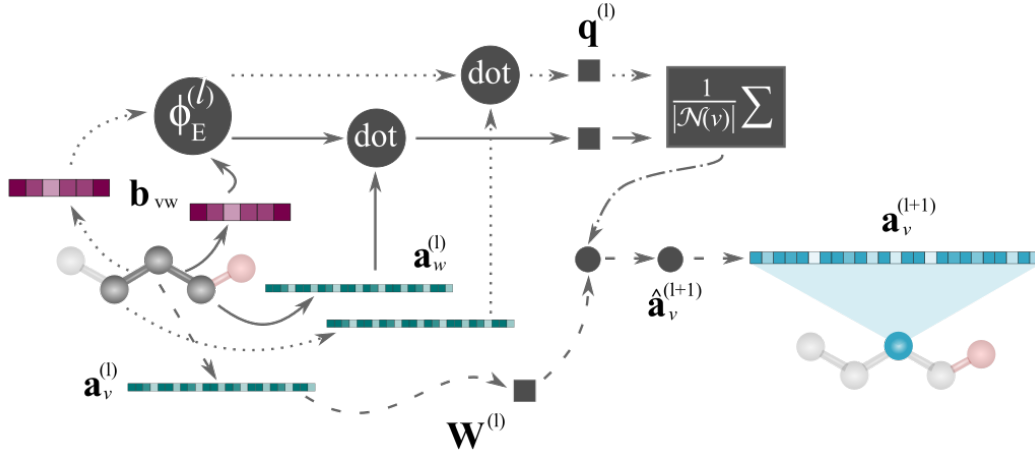


Fig. 3.3.: Schematic representation of the message-passing layer used for the prediction of isothermal IDACs.

$$\mathbf{a}_v^{(1)} = \phi_0(\mathbf{a}_v^{(0)}) \quad (3.1)$$

Then, for each message-passing layer l , the node embeddings of the corresponding graph are updated according to a message-passing mechanism similar to the generalization of the Gated Graph Neural Networks given by Gilmer et al. [52] and the edge-conditioned convolution proposed by Simonovsky et al. [148]:

$$\hat{\mathbf{a}}_v^{(l+1)} = \mathbf{W}^{(l)} \cdot \mathbf{a}_v^{(l)} + \frac{1}{|\mathcal{N}(v)|} \sum_{w \in \mathcal{N}(v)} \left(\phi_E^{(l)}(\mathbf{b}_{vw}) \cdot \mathbf{a}_w^{(l)} + \mathbf{q}^{(l)} \right) \quad (3.2)$$

Here, $\hat{\mathbf{a}}_v^{(l+1)}$ represents the updated $d^{(l+1)}$ -dimensional feature vector of node v in layer $l + 1$. The term $\mathbf{W}^{(l)} \in \mathbb{R}^{d^{(l+1)} \times d^{(l)}}$ refers to a learnable weight matrix that transforms the embedding of node v . This transformation is linear and is akin to the weights used in traditional neural networks for feature transformation. $\mathcal{N}(v)$ is the set of neighboring nodes for the specific node v , ensuring that messages are passed only between directly connected nodes. The function $\phi_E^{(l)}(\cdot)$, which processes the edge feature vector \mathbf{b}_{vw} between nodes v and $w \in \mathcal{N}(v)$, outputs an edge-conditioned weight matrix in $\mathbb{R}^{d^{(l+1)} \times d^{(l)}}$. In this work, $\phi_E^{(l)}(\cdot)$ is a single hidden-layer neural network employing the ReLU activation function. The primary objective of the neural network $\phi_E^{(l)}(\cdot)$ is to dynamically adjust the contribution of each neighbor's features to node v based on the characteristics of the edge connecting them. It is important to emphasize that the matrix generated by $\phi_E^{(l)}(\cdot)$ is dynamically generated

during the message-passing process. The actual parameters that are determined during training correspond to the parameters of $\phi_E^{(l)}(\cdot)$, the bias vector $\mathbf{q}^{(l)} \in \mathbb{R}^{d^{(l+1)}}$, and the matrix $\mathbf{W}^{(l)}$. The message-passing operations of layer l (Eq. 3.2) are schematically represented in Fig. 3.3.

After each message-passing layer l , batch normalization [73] is used to standardize the node embeddings across the training mini-batch, enhancing the stability and efficiency of the training process. Moreover, the Leaky ReLU activation function is used to introduce non-linearity into the framework. This can be written as follows

$$\mathbf{a}_v^{(l+1)} = \text{Leaky ReLU} \left(\frac{\hat{\mathbf{a}}_v^{(l+1)} - \mathbb{E}[\hat{\mathbf{a}}_v^{(l+1)}]}{\sqrt{\text{Var}[\hat{\mathbf{a}}_v^{(l+1)}] + \epsilon}} \odot \beta_1^{(l+1)} + \beta_2^{(l+1)} \right) \quad (3.3)$$

where $\mathbb{E}[\cdot]$ and $\text{Var}[\cdot]$ represent the expectation and variance operators, ϵ is a small number to avoid zero-division, \odot denotes element-wise multiplication and $\beta_1^{(l+1)}$ and $\beta_2^{(l+1)}$ are learnable vector parameters. These vector parameters are utilized to maintain the expressiveness of $\mathbf{a}_v^{(l+1)}$ by adjusting its output distribution dynamically.

Dropout [72] is used after each message-passing iteration to avoid overfitting by reducing the network's reliance on specific neurons. This is done by randomly setting a fraction of the vector $\mathbf{a}_v^{(l+1)}$ entries to 0 at each update during the training phase. After each message-passing layer l , an updated graph is obtained with the same connectivity and edge features, but with updated node embeddings according to Eqs. 3.2 and 3.3. The final set of node embeddings $\{\mathbf{a}_v^{(final)} | v \in \mathcal{V}\}$ is obtained by averaging the corresponding intermediate node embeddings obtained after each message-passing layer, as shown in Eq. 3.4. In Eq. 3.4, L denotes the total number of message-passing layers in the GNN. This process is known as *jumping knowledge* and has been reported [169] to improve the accuracy of GNN graph representation learning.

$$\mathbf{a}_v^{(final)} = \frac{1}{L} \sum_l \mathbf{a}_v^{(l)}; \quad \forall v \in \mathcal{V} \quad (3.4)$$

After this, the final set of node embeddings $\{\mathbf{a}_v^{(final)} | v \in \mathcal{V}\}$ is processed via the Set2Set [157] global pooling mechanism, yielding a vector that captures the information of the corresponding molecular species, serving as its (learned) molecular

fingerprint tailored for IDAC predictions. In this work, the same Set2Set pooling has been used for the updated solute and solvent graphs.

The Set2Set pooling transforms the set of final node embeddings iteratively using a Long Short-Term Memory (LSTM) network and the softmax function to compute attention weights according to the following

$$\mathbf{q}_t = LSTM(\mathbf{q}_{t-1}^*) \quad (3.5)$$

$$\alpha_{v,t} = \text{softmax}(\mathbf{a}_v^{(final)} \cdot \mathbf{q}_t) \quad (3.6)$$

$$\mathbf{r}_t = \sum_{v \in \mathcal{V}} \alpha_{v,t} \mathbf{a}_v^{(final)} \quad (3.7)$$

$$\mathbf{q}_t^* = \mathbf{q}_t \parallel \mathbf{r}_t \quad (3.8)$$

where, \mathbf{q}_{t-1}^* is the hidden state of the LSTM at processing step $t - 1$, $\alpha_{v,t}$ is the attention weight for node v at processing step t , \mathbf{r}_t is the readout vector at processing step t , and \parallel represents concatenation. In this work, 3 processing steps were used. Notice, that the same hidden state vector \mathbf{q}_t is used to compute the attention weights for every node in the graph, and, as a result, the permutation invariance of the pooling operation is kept. Therefore, after the Set2Set global pooling layer, a single vector \mathbf{a}_g is obtained that condenses the information of the corresponding molecular graph (cf. Eq. 3.9):

$$\mathbf{a}_g = \mathbf{q}_{t=3}^* = \text{Set2Set}(\{\mathbf{a}_v^{(final)} | v \in \mathcal{V}\}) \quad (3.9)$$

As previously mentioned, the vector \mathbf{a}_g for the solute and the solvent is computed by using a single Set2Set layer. This means that the learnable parameters of the LSTM network (Eq. 3.5) are the same despite of processing the solute or the solvent species. Hence, the global pooling operation is just tailored to compress the information of the final graphs independently of their composition in the mixture.

Finally, the binary mixture is represented by the concatenation of the solvent $\mathbf{a}_{g=\text{solvent}}$ and the solute $\mathbf{a}_{g=\text{solute}}$ embeddings. This results in a vector \mathbf{a}_{mix} which is twice the size of the species embedding:

$$\mathbf{a}_{mix} = \mathbf{a}_{g=\text{solvent}} \parallel \mathbf{a}_{g=\text{solute}} \quad (3.10)$$

This vector \mathbf{a}_{mix} is then fed to a multi-layer perceptron (MLP) that predicts the corresponding $\ln \gamma_i^\infty$ value. Batch normalization, the Leaky ReLU activation function, and dropout are also used in each hidden layer of the MLP. This entire GNN-based framework is represented in Fig. 3.4.

The main advantage of this approach, compared to the alternative phenomenological models, is that it can be trained end-to-end from the molecular structures to the IDAC values using backpropagation. Therefore, contrary to the development of, for instance, group contribution methods involving the handpicking of molecular groups based on expert knowledge, the processing of molecular structures is performed automatically by the GNN-based model depending on the atomic and bond characteristics of the molecules. As a result, and compared to group contribution methods, the labor-intensive tasks of molecular fragmentation and binary-interaction parameter fitting (which have been the domain of a select number of specialists in the field for many years) are effectively accomplished/replaced by the proposed framework in a single automatic step.

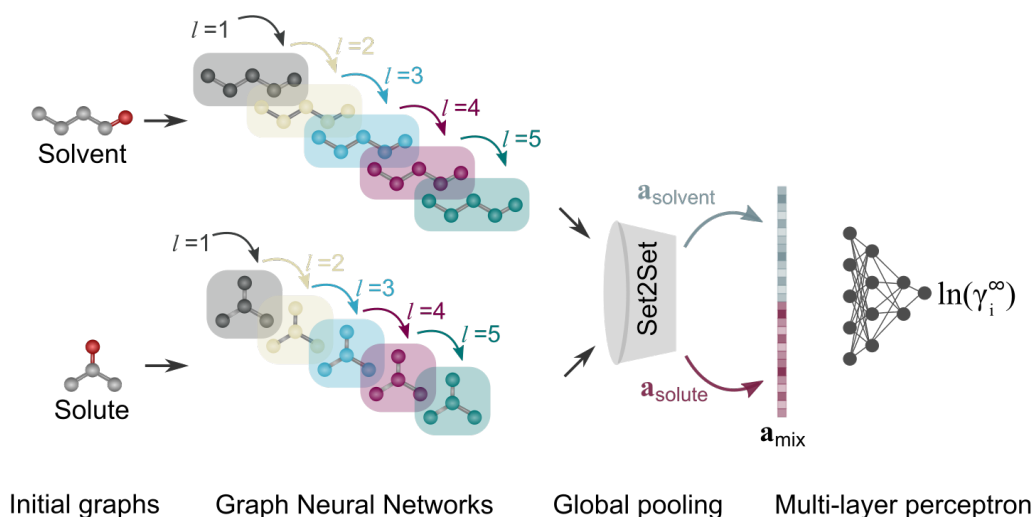


Fig. 3.4.: Schematic representation of the proposed GNN-based model for the prediction of isothermal IDACs.

The final MLP serves a dual purpose, it calculates the enthalpic contributions to non-ideality arising from intermolecular interactions between the solute and solvent, and it also assesses the entropic contributions resulting from disparities in molecular size and shape between the two. This computation leverages the binary-mixture representation, \mathbf{a}_{mix} , as a foundational input, enabling the MLP to integrate these complex physicochemical phenomena into the same analytical framework. This vectorial mixture representation is in contrast to the representation of mixtures based on a set of functional groups used in group contribution methods (e.g., UNIFAC).

3.3.1 Training

The hyperparameters of the proposed GNN-based model for predicting isothermal IDACs are given in Appendix A.8, along with the details on how they were determined via Bayesian optimization. Both GNNs used for processing the solvent and the solute used the same hyperparameters. Therefore, both networks can be regarded as identically structured, differing only by the actual model parameters determined during the training. The complete GNN-based model has 619,483 trainable parameters.

The Brouwer dataset described in Section 3.1 was divided into two sets: 80% allocated for model development and the remaining 20% designated for testing. 90% of the model development set was directly utilized for model training, and the remaining 10% served for validation and fine-tuning of hyperparameters. Since the Brouwer data set comprises only isothermal data, a random split was chosen to construct the train/validation/test splits. As further discussed in Chapter 4, the random data set splitting primarily evaluates the model's performance in interpolating across the solute-solvent chemical space delineated by the training data set. The evaluation of the model in this way mirrors the practical application of phenomenological models, which are employed within their specific domains of applicability, such as for compounds that are polar, aliphatic, or aromatic [28, 56]. This strategy ensures that the model's capabilities are assessed within contexts relevant to its intended use. This also simultaneously offers an advantage by suggesting an applicability domain for the GNN (constructed from the solute-solvent training space). The resulting model development and test sets follow a similar distribution of $\ln(\gamma_i^\infty)$ values covering the whole range of available values in the data set (cf. Appendix A.9).

The proposed GNN-model was developed in Python 3.8, using the PyTorch [116] and PyTorch Geometric [46] libraries for its implementation. The training was carried out using the Adam algorithm, with the mean squared error (MSE) as the loss function. A learning rate adjustment mechanism was implemented, decreasing the rate by 0.8 when the validation loss failed to improve beyond a threshold of 10^{-4} for three successive epochs. All computational experiments were carried out in a single NVIDIA Tesla P100 GPU (16 GB).

3.3.2 Ensemble learning

Ensemble learning refers to the combination of multiple models (in this case, multiple GNN models) to improve the robustness and accuracy of predictions. In this

approach, several GNN models, each with varied random train/validation splits, are trained independently to predict isothermal IDACs. The individual predictions from these models are then aggregated, through averaging, to produce a final, unified prediction. This technique utilizes the chemical diversity learned among GNNs to reduce the impact of any single GNN's biases or errors, thereby enhancing the overall predictive performance. The use of ensemble learning usually results in better generalization to unseen data.

The ensemble size in this work was established through a systematic evaluation of how the inclusion of additional GNN models, based on their performance in the training and validation sets, impacts the ensemble's overall performance. Fig. 3.5 illustrates the incremental progression of mean absolute percentage error (MAPE) as the ensemble expands. Notably, the MAPE begins to plateau after the inclusion of approximately 15 models. To reinforce the stability and reliability of the predictions, and to enhance the statistical significance of the standard deviation given for a single GNN model (cf. Table 3.3), a total of 30 models were ultimately integrated into the ensemble. The smoothed tendency line shown in Fig. 3.5 was calculated as the centered moving average with a window size of 7 and a minimum window of 1.

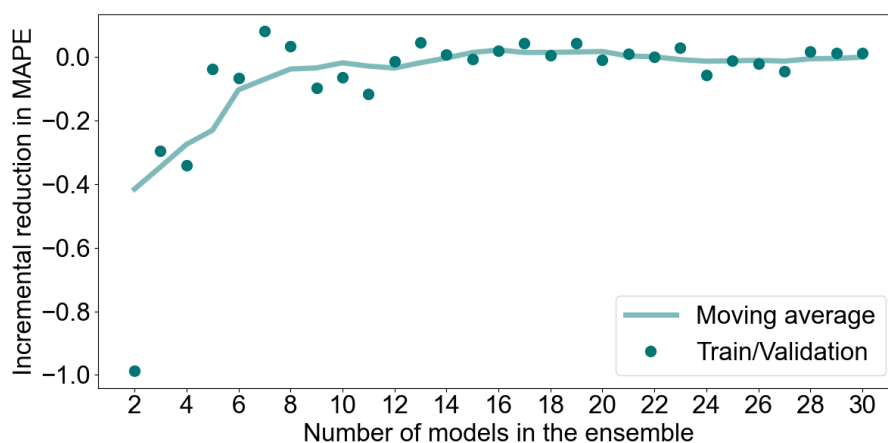


Fig. 3.5.: Incremental performance in the mean absolute percentage error (MAPE) with respect to the ensemble size of the proposed GNN-based for isothermal IDACs.

3.3.3 Comparison to phenomenological models

Table 3.3 showcases the comparative performance of the proposed GNN model, with eight of the most popular IDAC phenomenological models. A baseline comparison is established with the Hildebrand model. Beyond this baseline, the remaining models exhibit significantly enhanced performance. Except for the HSP model,

these models are considered to represent the state-of-the-art in accurately predicting IDACs in real-world applications. Within Table 3.3, the standard deviation of the GNN performance is illustrated in between parenthesis. This standard deviation was computed across 30 distinct runs of the GNN model, each with varying random seed used for the train/validation splitting. The ensemble prediction (denoted by e-GNN in Table 3.3) is obtained by averaging the predictions of the 30 individual GNN models.

Tab. 3.3.: Performance comparison between the proposed GNN model and popular phenomenological models for isothermal IDAC prediction.

Model	Coverage \uparrow	MAE \downarrow	SDAE \downarrow	R ² \uparrow	MAPE \downarrow
Hildebrand	54.66%	2.55×10^5	9.89×10^6	-7.92×10^9	4.26×10^5
HSP	56.23%	15.86	122.23	-0.27	66.90
UNIFAC (Ly)	93.91%	10.32	59.17	0.56	32.65
UNIFAC	94.52%	10.44	60.39	0.54	32.06
COSMO-RS	97.22%	10.64	66.98	0.43	28.37
UNIFAC (Do)	94.91%	8.23	56.50	0.60	25.88
Abraham	44.27%	4.16	33.58	0.90	21.93
MOSCED	45.69%	2.78	12.58	0.48	19.88
GNN (test)	100%	4.36(0.33)	30.49(2.79)	0.77(0.04)	24.94(2.07)
GNN (all)	100%	4.12(0.36)	31.26(5.44)	0.87(0.05)	16.22(0.90)
e-GNN (test)	100%	3.61	26.49	0.83	22.41
e-GNN (all)	100%	3.30	24.12	0.92	14.05

Note: Performance metrics include Mean Absolute Error (MAE), Standard Deviation of Absolute Errors (SDAE), Coefficient of Determination (R²), and Mean Absolute Percentage Error (MAPE), calculated using unscaled $\gamma_{i,j}^\infty$ values. The arrows in the metric names indicate whether a larger or smaller value is better.

The Standard Deviation of Absolute Errors (SDAE), shown in Table 3.3, is calculated as

$$\text{SDAE} = \sqrt{\frac{\sum_{i=1}^{n_D} (r_i - \mu_r)^2}{n_D}} \quad (3.11)$$

$$\mu_r = \frac{\sum_{i=1}^{n_D} r_i}{n_D} \quad (3.12)$$

$$r_i = |\gamma_i^\infty - \hat{\gamma}_i^\infty| \quad (3.13)$$

where, n_D refers to the number of data points in the data set and $\hat{\gamma}_i^\infty$ refers to the predicted IDAC value.

A critical distinction between the GNN models and the traditional phenomenological approaches lies in the scope of data coverage. The GNN models have the capability to predict across the entirety of the data set, in contrast to the shown phenomenological

models which are confined to narrower chemical spaces. For example, the applicability of UNIFAC-based models is limited to scenarios where binary-interaction parameters for the relevant UNIFAC groups are known. The availability of parameters similarly constrains the predictive reach of the other models. The exception in this context is COSMO-RS, whose application depends upon the accessibility of Density Functional Theory (DFT) calculations for the mixture constituents. In this study, DFT calculations were available for 97.22% of the cases, slightly limiting the COSMO-RS model’s coverage. Therefore, based on the dependency of component parameters, models like Hildebrand, Hansen, MOSCED and Abraham are usually not considered truly predictive.

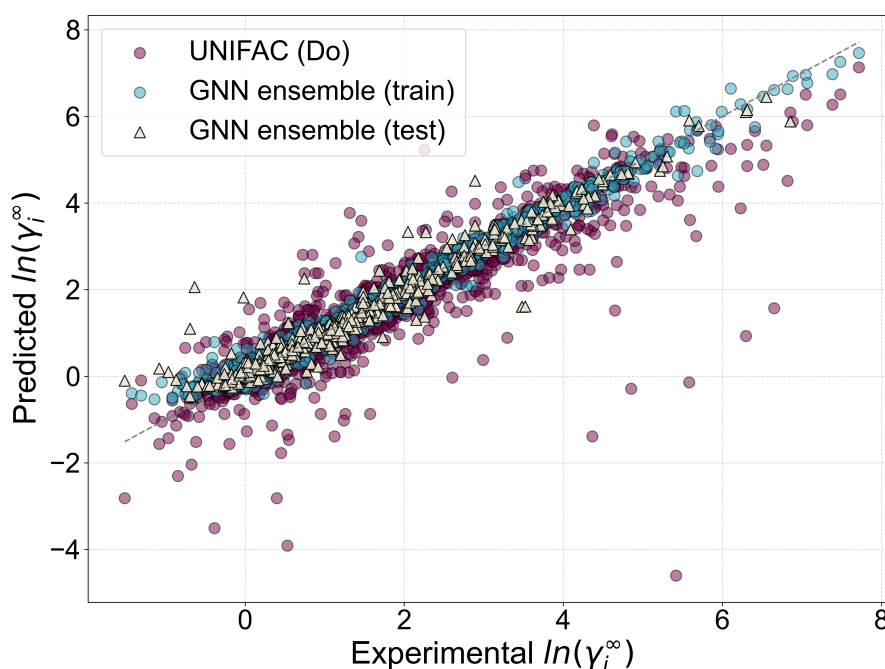


Fig. 3.6.: Parity plot between the experimental and the predicted IDAC values by GNN ensemble and UNIFAC-Dortmund. All the feasible systems for each method are shown.

Considering the varying degrees of data set coverage by different models as presented in Table 3.3, comparing their performance is indeed far from straightforward. The eight phenomenological models’ performance is detailed for their entire feasible data set, whereas the GNN’s performance is reported for the test set. Additionally, for a holistic view, the GNN’s performance over the entire data set, denoted with “(all)” in Table 3.3, is also provided. Focusing on the test set, the GNN model demonstrates a lower MAE compared to most phenomenological models, with the notable exceptions being the Abraham and MOSCED approaches. However, these

two models exhibit limited system coverage, 44.27% and 45.69% respectively, in contrast to the GNN's complete coverage.

When narrowing the comparison to models with a data coverage of over 90% (i.e., UNIFAC-based, COSMO-RS and GNN-based models), the GNNs surpass them across all metrics. Yet, it is observed that the MAPE for UNIFAC-Dortmund is within one standard deviation of the GNN's performance, suggesting a comparable effectiveness according to this metric. However, the GNN's significantly lower MAE (cf. Table 3.3) suggests that the GNN is able to predict highly non-ideal systems (where IDAC values are larger) considerably better than UNIFAC-Dortmund. This can be observed in the parity plot shown in Fig. 3.6, where for IDAC values exceeding 4, the e-GNN model's predictions more closely mirror the actual experimental values. However, also an important limitation can be observed when looking at systems with the lowest IDACs. In such cases, the GNN systematically overestimates the IDAC values. This might be a result of the relatively few systems in such range contained in the data set (cf. IDAC value distributions in Appendix A.9). It is important to highlight that Fig. 3.6 encompasses all systems considered feasible by both the e-GNN and UNIFAC-Dortmund methods. The e-GNN model outperforms all UNIFAC-based models and COSMO-RS across all metrics.

If one considers the performance of the GNN models (both the ensemble and the single GNN), their SDAE is only outperformed by the MOSCED model. This suggests that the isothermal IDAC predictions of GNN-based models are more consistent compared to most phenomenological models. In other words, GNN predictions are uniformly closer to the true values, and contain fewer outliers or extreme errors compared to the rest of the models (except for MOSCED). The relatively low SDAE also suggest that the GNNs could be more trustworthy, at least in the tested chemical space, compared to most phenomenological models given that the error is less likely to vary widely from one prediction to the next. By looking now at the R^2 values, the GNN-based models outperformed the rest of the phenomenological models, except for the Abraham model. This suggests that the GNN can indeed capture the underlying pattern between the molecular structures and the corresponding IDAC value across the data set better than almost all phenomenological models.

Thus, it is interesting to consider the relatively good performance of the GNN compared to well-established phenomenological models. This, of course, needs to be considered through the lens of the specific chemical space studied here. But, perhaps even more intriguing, one has to consider the fact that the Abraham and MOSCED models perform comparatively well or even better than the GNN model. This raises questions about the reasons behind their success and the specific

type of information these models use to attain such results. On one hand, it is expected that models tailored for a constrained chemical space (such as Abraham and MOSCED, characterized by lower data coverage, cf. Table 3.3), would exhibit strong predictive accuracy for those specific mixtures. The reason for this is that models optimized within a narrower chemical domain are inherently designed to address fewer types of physical phenomena than might be encountered in a broader context. On the other hand, a defining characteristic that sets the Abraham and MOSCED models apart from other phenomenological models is their inclusion of model parameters specifically designed to address various van der Waals and hydrogen-bonding intermolecular interactions (as elaborated upon in Appendices A.5 and A.7).

3.3.4 Robustness analysis with 5-fold cross-validation

In order to test the robustness of the proposed GNN-based framework for predicting isothermal IDACs, 5-fold cross-validation was used. This helps ensuring that the assessment of the GNN model is not dependent on a particular random split of the training and test data. This also ensures that all data is used for testing the model. Table 3.4 shows the comparative performance of the proposed GNN model and each of the phenomenological models presented before. The Hildebrand model is excluded here due to its poor performance for predicting IDACs (cf. Table 3.3). In general, the GNN model outperforms all phenomenological models when considering the mean value across all range of metrics. Two exceptions are noted with the Abraham model, which outperforms the GNN in MAE, standard deviation of absolute errors (SDAE), and coefficient of determination (R^2), and the MOSCED model which outperforms the GNN in terms of the MAPE.

Regarding prediction variability, as indicated by the standard deviation values presented in between parentheses in Table 3.4, the GNN model demonstrates greater consistency than all phenomenological models with respect to both MAE and the SDAE. On the other hand, for R^2 and MAPE, the GNN model tends to exhibit higher variability. This observation is attributed to the use of the MSE as a loss function during the GNN training. Since, the MSE penalizes higher absolute errors more than small absolute errors, the GNN is lead during training to minimize absolute errors rather than relative percentage errors. Despite the GNN model's increased variability in R^2 and MAPE, the assessment of the model's overall robustness should consider both mean and standard deviation values. For example, as illustrated in Fig. 3.7 with the MAPE, despite a wider variability in the percentage errors of GNN predictions relative to those from phenomenological models, the GNN predictions

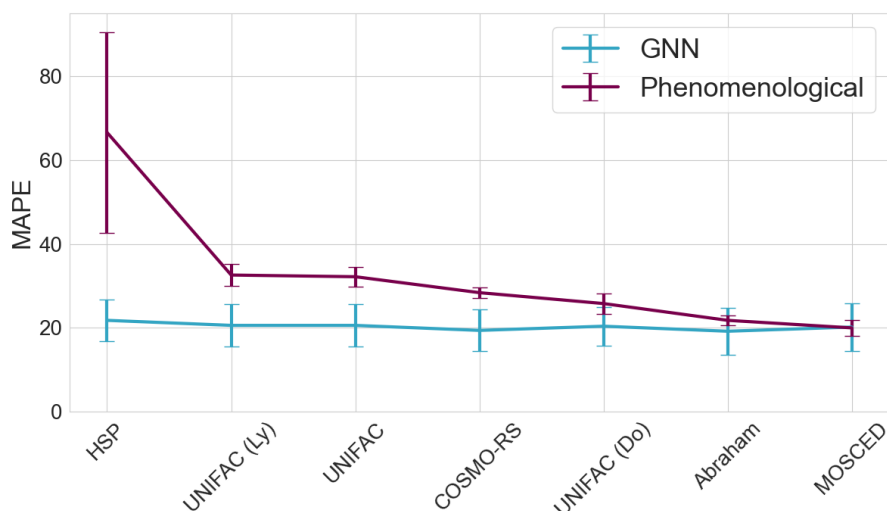


Fig. 3.7.: Mean absolute percentage error (MAPE) for the proposed GNN and popular phenomenological models when predicting IDACs at 298.15 K. Results are shown only for feasible systems of the corresponding phenomenological model contained in the GNN test set during a 5-fold cross-validation.

Tab. 3.4.: Performance comparison between the proposed GNN model and popular phenomenological models using 5-fold cross validation when predicting IDACs at 298.15 K.

Model	MAE ↓	SDAE ↓	R ² ↑	MAPE ↓
HSP	15.85 (5.64)	102.42 (66.99)	-0.49 (1.20)	66.6 (23.89)
GNN (test)	5.28 (0.84)	34.64 (12.47)	0.75 (0.18)	21.8 (5.04)
UNIFAC (Ly)	10.31 (1.82)	55.05 (21.03)	0.48 (0.16)	32.6 (2.58)
GNN (test)	4.98 (1.09)	35.92 (14.22)	0.69 (0.22)	20.6 (5.00)
UNIFAC	10.43 (1.99)	56.04 (21.91)	0.48 (0.13)	32.2 (2.40)
GNN (test)	5.00 (1.10)	35.83 (14.19)	0.69 (0.22)	20.6 (5.00)
COSMO-RS	10.65 (2.64)	61.43 (26.83)	0.41 (0.05)	28.4 (1.20)
GNN (test)	4.98 (1.06)	35.40 (14.04)	0.69 (0.22)	19.4 (5.00)
UNIFAC (Do)	8.22 (1.84)	52.84 (19.63)	0.53 (0.11)	25.8 (2.48)
GNN (test)	4.64 (1.13)	32.49 (13.23)	0.74 (0.20)	20.4 (4.63)
Abraham	4.18 (1.18)	27.22 (19.68)	0.91 (0.06)	21.8 (1.17)
GNN (test)	4.41 (0.72)	31.39 (9.79)	0.8 (0.13)	19.2 (5.56)
MOSCED	2.78 (0.53)	12.15 (3.58)	0.47 (0.22)	20.00 (1.90)
GNN (test)	1.44 (0.49)	4.38 (2.53)	0.93 (0.05)	20.20 (5.74)

Note: Performance metrics include Mean Absolute Error (MAE), Standard Deviation of Absolute Errors (SDAE), Coefficient of Determination (R²), and Mean Absolute Percentage Error (MAPE), calculated using unscaled $\gamma_{i,j}^{\infty}$ values. The arrows in the metric names indicate whether a larger or smaller value is better. All metrics are given for the feasible systems of the corresponding phenomenological model contained in the test set of the GNN. The standard deviation across the 5-folds is given in between parenthesis. The best mean value is marked in bold.

generally outperform, except in the case of the chemically narrow Abraham and MOSCED models.

3.4 Hybrid parallel graph neural networks

In this Section, the integration of phenomenological IDAC models with the proposed GNN-based model is explored through a parallel arrangement. In the context of chemical processes, parallel hybrid models have shown notable improvements in accuracy and interpretability compared to the individual submodels alone [159, 54, 103]. Specifically, the parallel hybrid arrangement involves training a machine learning model to learn the errors generated by a physics-based model. The prediction of the hybrid model is constructed by adding the predictions of the submodels, thereby reducing the overall prediction error. This type of hybrid model arrangement is usually favored, against a sequential arrangement, for its implementation flexibility, allowing for the effective combination of phenomenological understanding and data-driven information to achieve superior predictive capabilities.

A hybrid parallel GNN model has been constructed for each of the eight phenomenological models presented in the previous Section. The construction procedure consists of three sequential steps. First, the corresponding phenomenological model is used for predicting the IDACs of the systems contained in the training set. Second, the residual $r_i = |\ln \gamma_i^\infty - \ln \gamma_i^{\infty,phe}|$ between the phenomenological model IDAC prediction and the actual IDAC value is calculated for each system in the training set. Third, the proposed GNN-based model (as described in the previous Section) is trained for predicting r_i (in contrast of the previous approach of predicting the IDAC value directly) from the corresponding solute-solvent molecular structures. Since, as shown in Table 3.3 and discussed in the previous Section, the performance of the ensemble of GNNs outperforms that of a single GNN, the ensemble of GNNs is here used for predicting r_i . The final IDAC prediction of the hybrid parallel GNN model can be computed as

$$\ln \gamma_i^{\infty,hyb} = \ln \gamma_i^{\infty,phe} + K r_i \quad (3.14)$$

where, the superscripts *hyb* and *phe* refer to the predictions of the hybrid and the phenomenological models, respectively. The parameter $K \in [0, 1]$ serves as a weighting factor for the correction term computed by the GNN model. This weighting factor has been set to one in the present study (implying a full application of the GNN correction). However, the application of K allows for tailored adjustments based on the GNN model's specific applicability domain. For instances where a prediction pertains to a system outside the GNN's applicability domain, K could be adjusted to zero, effectively nullifying the GNN correction and keeping a phenomenological "backbone" prediction. Moreover, a more elaborated approach could involve varying

K between zero and one. This variation would ideally depend on the confidence in the GNN's residual prediction. Such mechanism offers a dynamic tool for fine-tuning the influence of the GNN correction, enabling a balance between data-driven information and phenomenological predictions.

The evaluation of each hybrid parallel GNN model involved a comparative analysis against its corresponding phenomenological model as well as against a GNN ensemble model that directly predicts IDACs (i.e., the direct approach). To ensure consistency and fairness in comparison, identical hyperparameters and train/valid/test splits were applied across both the hybrid and direct approaches. Furthermore, to maintain fair conditions for assessment, the data set utilized for training and testing the models was precisely limited to the data points deemed feasible by each of the eight phenomenological models. This approach guarantees that the performance comparisons are based on a common ground, allowing for an accurate assessment of the effectiveness of the hybridization process relative to both traditional phenomenological models and standalone GNN models.

Tab. 3.5.: Performance comparison between the corresponding hybrid parallel GNN, the GNN ensemble (e-GNN) and the phenomenological models. Results are shown for the systems in the test set. The best value for each method is shown in bold.

Model	% points with $AE \leq 0.2$ \uparrow			# points with $AE \geq 1$ \downarrow		
	Phenom.	e-GNN	Hybrid GNN	Phenom.	e-GNN	Hybrid GNN
UNIFAC-Ly	39.96	78.41	74.62	56	5	5
UNIFAC	44.55	76.50	75.00	46	5	10
UNIFAC-Do	62.36	76.40	74.91	21	4	8
COSMO-RS	51.55	76.42	86.84	24	3	2
Abraham	62.65	74.70	83.13	1	4	1
MOSCED	70.04	73.93	85.99	4	2	1

Table 3.5 summarizes the test performance of the hybrid approaches compared to the stand alone submodels in terms of the percentage of points that have an absolute error in the logarithmic IDAC below 0.2 (% $AE \leq 0.2$), and the number of points that have an absolute error in the logarithmic IDAC above 1 (# points with $AE \geq 1$).

Figures 3.8 to 3.13 show the absolute error density between the predicted logarithmic IDAC values with respect to the experimental IDAC values. The prediction performance is shown for the corresponding phenomenological model, the GNN ensemble and the corresponding hybrid parallel GNN model. For visualization purposes, only errors in the range $(-1.5, 1.5)$ are shown. The number of test systems included in each case is also shown in the corresponding graph. It can be seen that the distribution of absolute errors for hybrid parallel GNN models generally clusters closer to zero in comparison to their constituent models (i.e, the phenomenological and pure GNN approaches), except for the UNIFAC-based models. This implies a

superior ability of the GNN to predict residuals of the non-UNIFAC phenomenological models more effectively than they do in predicting IDACs directly. A plausible explanation for this enhanced predictive performance lies in the inductive bias introduced by the phenomenological models, which positively influences the overall prediction accuracy of the hybrid framework. This also shows the presence of systematic errors inherent to the phenomenological models, errors which are overlooked by them, yet can be discerned and corrected by the GNN from the solute-solvent molecular structure information.

In the following, each hybrid parallel model's performance is discussed, comparing it to that of its component submodels. The Hildebrand and HSP models have been omitted from this specific discussion because their performance falls short when measured against the other phenomenological models, and the GNN-based models clearly outperform them.

3.4.1 Hybrid UNIFAC-based GNN models

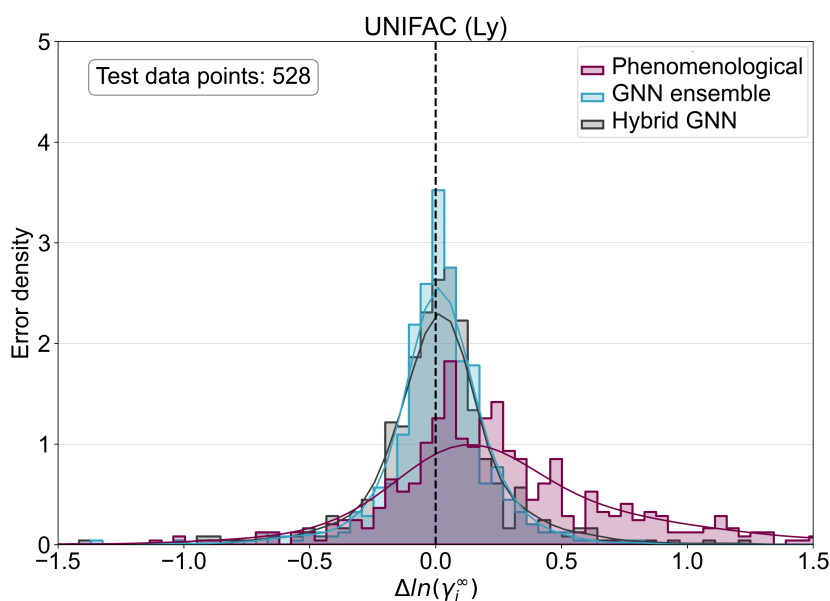


Fig. 3.8.: Absolute error density of UNIFAC (Lyngby), the GNN ensemble and the corresponding hybrid GNN model. The black central line shows the null-error for comparison. The results are shown for systems in the test set.

An initial observation reveals that the predictions generated by the UNIFAC-Lyngby (cf. Fig. 3.8) and UNIFAC (cf. 3.9) models exhibit a tendency towards positive deviations from the true values. This trend can be explained from the models' need to extrapolate beyond the finite concentration data utilized during their development,

extending into the infinite dilution regime. This extrapolation challenge arises because the parameters for these UNIFAC models were calibrated exclusively against vapor-liquid equilibria (VLE) and liquid-liquid equilibria (LLE) data [47]. The Lyngby modification to UNIFAC incorporated only additional excess enthalpy data alongside VLE and LLE data [95] for its parameter fitting, but not IDAC data. In contrast, the GNN model demonstrates, in both cases (cf. Fig. 3.8 and Fig. 3.9), a capacity to predict a broader spectrum of systems with errors more closely aligned to zero.

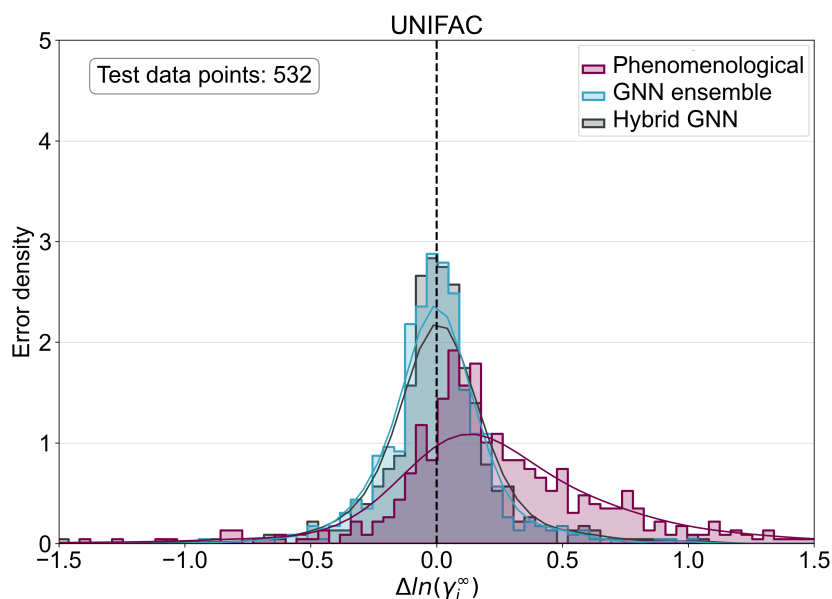


Fig. 3.9.: Absolute error density of UNIFAC, the GNN ensemble and the corresponding hybrid GNN model. The black central line shows the null-error for comparison. The results are shown for systems in the test set.

Nevertheless, it is interesting to note that the performance of the hybrid GNN model, when applied in conjunction with both UNIFAC-Lyngby and UNIFAC models, slightly underperforms relative to the GNN model trained directly on IDAC experimental data. This pattern follows the observations reported before when discussing the performance of hybrid matrix completion methods coupled with the UNIFAC-Dortmund model [75]. But, despite the GNN model slightly outperforming the hybrid GNN, the latter still effectively reduces the errors inherent to the phenomenological models, aligning the majority of absolute errors closer to zero.

It is evident from Fig. 3.10, that the UNIFAC-Dortmund model demonstrates a superior ability to predict IDACs when compared to both the UNIFAC and UNIFAC-Lyngby models. This enhanced performance is anticipated, considering that the development of this model variant explicitly incorporated experimental IDAC data

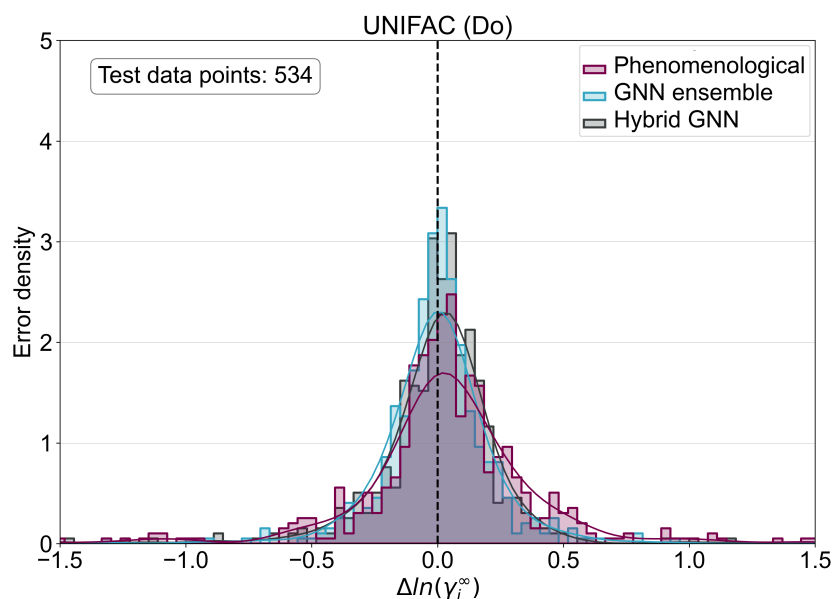


Fig. 3.10.: Absolute error density of UNIFAC (Dortmund), the GNN ensemble and the corresponding hybrid GNN model. The black central line shows the null-error for comparison. The results are shown for systems in the test set.

into the calibration of its parameters [59]. Despite this improvement, the GNN model continues to outperform, delivering more accurate predictions across a wider array of systems. The gap between the GNN and the hybrid GNN model performances narrows in this context, with the GNN model, exhibiting predictions that are marginally more aligned towards zero. This observation is again aligning with the discussions in the literature [75].

Hence, it becomes apparent that the effectiveness of data-driven models, particularly when tasked with predicting the residuals of mechanistic or phenomenological models, is significantly affected by the underlying structures of these models [119]. Large systematic deviations can confound the data-driven component, rendering it more challenging for the model to accurately learn the residuals as opposed to learning the properties directly. For UNIFAC-based models, the presence of numerous severe outlier predictions (cf. Table 3.5) could account for their relatively negative impact on the performance of hybrid GNN models. This phenomenon has been similarly highlighted by Jirasek et al. [75].

3.4.2 Hybrid COSMO-RS GNN model

In the case of the hybrid GNN model derived from COSMO-RS predictions, a clear pattern can be observed: the hybridization strategy yields more accurate predictions

compared to either the GNN or COSMO-RS models independently (cf. Fig. 3.11). This observation is particularly interesting, especially when contrasted with the earlier discussion regarding UNIFAC-based models, which inherently integrate a stronger data-driven component through molecular fragmentation into functional groups and subsequent group parametrization compared to the more theoretical foundation of COSMO-RS. This underscores the potential advantage of combining insights coming from different sources (e.g., quantum chemistry and experimental IDAC values correlated to molecular structure). Such a synergistic integration of fundamentally different types of information is less explicitly realized in the UNIFAC-based implementations.

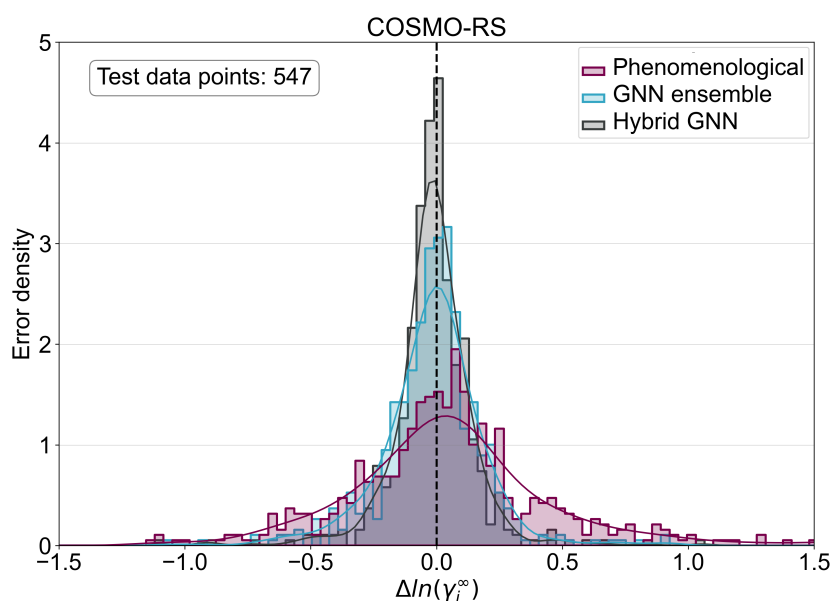


Fig. 3.11.: Absolute error density of COSMO-RS the GNN ensemble and the corresponding hybrid GNN model. The black central line shows the null-error for comparison. The results are shown for systems in the test set.

The observed improvements in prediction robustness by the hybrid GNN model, especially in enhancing the COSMO-RS model's performance beyond that of, for example, the UNIFAC-Dortmund model, are elucidated in Table 3.5. While COSMO-RS alone can only predict 51.55% of the data points (i.e., 282 points) relatively well, compared to UNIFAC-Dortmund which can predict 62.36% (i.e., 333 points) well, the hybridization of COSMO-RS elevates the proportion of accurately predicted points to 86.84%, as opposed to 74.91% for its UNIFAC-Dortmund counterpart. Notably, despite both COSMO-RS and UNIFAC-Dortmund exhibiting similar frequencies of significant outlier predictions errors (cf. 21 and 24 data points respectively in Table 3.5), the hybrid model built upon COSMO-RS significantly outperforms its UNIFAC-Dortmund equivalent in minimizing the occurrence of such inaccuracies.

3.4.3 Hybrid Abraham and MOSCED GNN models

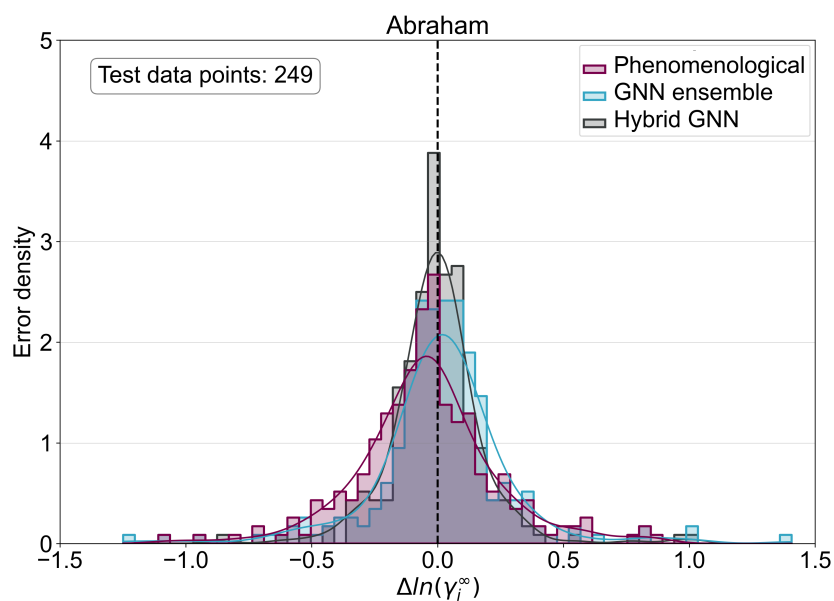


Fig. 3.12.: Absolute error density of Abraham, the GNN ensemble and the corresponding hybrid GNN model. The black central line shows the null-error for comparison. The results are shown for systems in the test set.

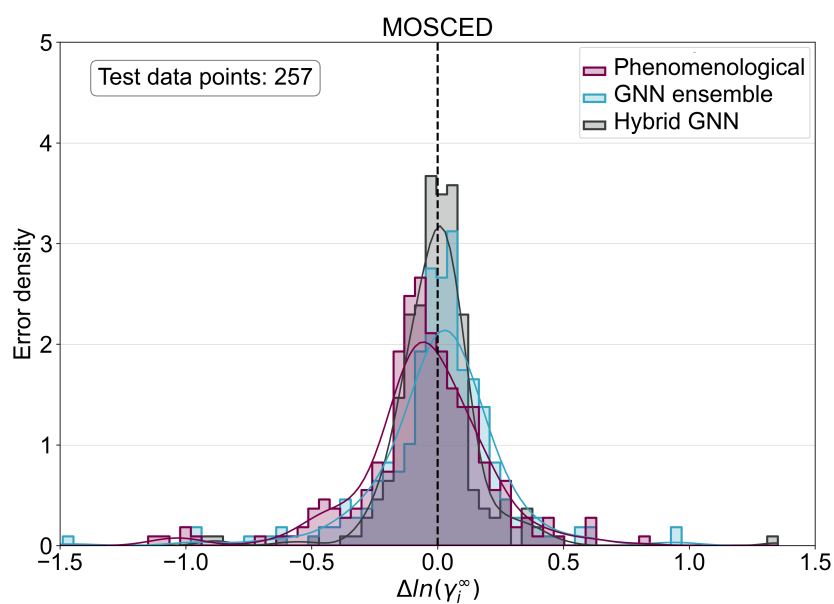


Fig. 3.13.: Absolute error density of MOSCED, the GNN ensemble and the corresponding hybrid GNN model. The black central line shows the null-error for comparison. The results are shown for systems in the test set.

Figures 3.12 and 3.13 illustrate comparable performances for the Abraham and MOSCED models, particularly noting their tendency towards negative deviations

from the true IDAC values. In both instances, employing the hybrid GNN methodology enhances the robustness and overall accuracy of the predictions beyond what is achievable by the standalone submodels. Furthermore, the performance gap between the phenomenological models and the GNN ensemble remains minimal, aligning with the insights presented in earlier discussions in Section 3.3.

3.5 Additional isothermal studies

In this Section, the previous discussion around the prediction of isothermal IDACs using GNNs is extended to multiple temperatures other than 298.15 K. Moreover, in this extended analysis, a more chemically diverse data set is used aiming to enhance the robustness of the observations discussed in previous Sections regarding the suitability of GNN-based models for predicting isothermal IDACs compared to phenomenological models. Particular attention is paid to a comparison with the COSMO-RS, UNIFAC-Dortmund, and MOSCED models, which were identified (cf. Table 3.3) as exhibiting superior performance at 298.15 K. This superior performance is noted not only in terms of the predictions' accuracy (especially notable with MOSCED), but also in terms of having a broad spectrum of feasible systems they can predict (a strength of both COSMO-RS and UNIFAC-Dortmund). Other phenomenological models are not considered in this comparison due to their relatively lower performance, both in terms of prediction accuracy and/or applicability range. The Abraham model is omitted due to its performance, which, albeit close, is slightly inferior to that of MOSCED. Thus, this extended isothermal comparative analysis includes one specialized and highly accurate model (MOSCED) and two models (COSMO-RS and UNIFAC-Dortmund) known for their wide applicability across various system types.

3.5.1 Data preprocessing

For these additional isothermal studies, the data set utilized originates from the DECHEMA Chemistry Data Series Vol. IX [55], one of the most comprehensive experimental data collections for IDACs currently available. The employment of this DECHEMA database, which is contained in physical books, as opposed to similar data collections in digital versions (e.g., Dortmund Data Bank [41]) facilitates the open-sourcing of models derived from it, such as those presented in this dissertation. Making the developed tools universally accessible to researchers and engineers worldwide, regardless of their financial resources, could significantly contribute to

the overall benefit of society. The physical volumes of the DECHEMA collection have been digitized using Optical Character Recognition (OCR) techniques. This was followed by extensive manual verification to ensure the accuracy and reliability of the digital content. Errors found in the original DECHEMA Chemistry Data Series Vol. IX [55] collection are reported in Appendix A.10.

The data contained in the DECHEMA Chemistry Data Series Vol. IX [55] encompasses IDAC values for binary systems, measured through various experimental techniques including gas-liquid chromatography, solubility measurements, static methods, ebulliometry, and among others. An in-depth review by [39] provides further insights into these experimental methodologies. Appendix A.11 includes further details on the experimental techniques used to collect all available IDACs and the relative distribution of data points collected by each experimental method. For the purposes of this extended isothermal analysis only systems containing organic molecules were included.

Regarding experimental uncertainty, it is notable that many scientific publications on IDAC measurements do not specify the method uncertainties. Nevertheless, literature sources, including [33], offer general estimations of absolute experimental IDAC uncertainties ranging from 0.1 to 0.2 (for the logarithmic IDACs), with relative uncertainties between 1% and 6% reported by other researchers [100, 40, 160, 102]. Brouwer et al. [26] estimate a minimum relative uncertainty of 5% for their data set, aligning with other uncertainty estimations found in literature [10, 39]. The discussion on model prediction accuracy in these additional isothermal studies is framed within the context of these experimental uncertainty estimations, specially with respect to the one reported for absolute values of $\ln \gamma_i^\infty$ [33].

In the process of compiling the data set from DECHEMA Chemistry Data Series Vol. IX [55], measurements of the same binary systems at identical temperatures were averaged to derive a single value per system at that specific temperature. The data set was refined by excluding compounds with ambiguous SMILES identifications, such as certain commercial solvents. This rigorous selection process resulted in a comprehensive data set, referred to as the *DECHEMA data set*, which encompasses 40,216 data points spanning 866 solvents and 1,032 solutes, with a total of 1,576 distinct compounds. Out of all 2,482,200 possible solute-solvent combinations, only 0.59% (14,663 binary systems) are reported in this data set. This highlights the extreme sparsity of the measured data even when only considering the solute-solvent space defined by the 1,576 distinct compounds contained in the data set.

Despite the sparsity of the measured data, the DECHEMA data set stands out for its coverage in terms of both the number of chemical species and experimental data

Tab. 3.6.: Comparison of the DECHEMA data set against similar IDAC data sets used in the literature.

# solutes	# solvents	# observed binary systems	Reference
378	414	7107	[33]
373	349	6416	[166]
295	407	7668	[151]
1,032	866	14,663	This work

points, surpassing data sets utilized in recent machine learning studies focused on matrix completion methods [33, 151] and natural language processing [166]. Table 3.6 shows the specific comparison in terms of the number of distinct solutes and solvents, and the number of observed binary systems.

The relatively larger number of distinct solutes and solvents available in the DECHEMA data set (cf. Table 3.6) allows for the analysis of the developed models in a much chemically diverse setting. This would potentially, not only increase the capacity of the models to learn the relationship between IDACs and molecular graphs, but also increase the robustness of the applicability domain of the models, revealing potential limitations that, perhaps, could be missed if the analysis would be performed in a much limited data set. Temperatures in the DECHEMA data set go from 213.15 to 562.45 K, with 90% of the data concentrated between 293.15 and 393.15 K. Furthermore, among the 14,663 observed binary systems, only 6,326 systems had their temperatures measured across a range of at least 20 K. The data set reveals diverse chemical behavior, with 22.28% of the data points exhibiting negative deviations from ideality, 77.31% showing positive deviations, and a marginal 0.41% approaching ideality.

Once the data was cleaned, a stratified splitting technique was adopted to define the training and test data sets. Initially, every compound was categorized based on the Classyfire ontology [38], resulting in the organization of the 1,576 distinct chemical compounds from the DECHEMA data set into 91 unique chemical classes. Predominantly, these included “benzene and substituted derivatives” with 270 compounds and “organooxygen compounds” with 193 compounds. A total of 17 compounds were not able to be classified into a specific class by Classyfire [38], these were grouped together for the purposes of the splitting. A more detailed enumeration of these chemical classes along with the corresponding compound count is presented in Appendix A.11. When considering these 91 chemical classes, the data points present in the DECHEMA data set can be grouped into 841 distinct binary combinations of chemical classes. For each of these combinations, we applied an 80/20 random split to establish the training and test data sets. In scenarios where a combination comprised solely a single solute-solvent pair, it was automatically allocated to the

Tab. 3.7.: Information of the isothermal subsets obtained from the DECHEMA data set and used for the extended isothermal study.

T (K)	# solutes	# solvents	Size of matrix	# obs.	% obs.	% train	% test
293.15	333	118	39,294	1,548	3.94	79	21
298.15	488	239	116,632	3,719	3.19	79	21
303.15	364	321	116,844	3,812	3.26	80	20
313.15	376	196	73,696	2,458	3.34	80	20
323.15	413	189	78,057	2,687	3.44	81	19
333.15	408	220	89,760	2,762	3.08	79	21
343.15	400	119	47,600	1,852	3.89	80	20
353.15	412	127	52,324	1,746	3.34	78	22
373.15	266	108	28,728	1,323	4.61	79	21

training set. This stratified splitting approach ensures that the models developed are exposed during training to a diverse set of distinct types of molecular interactions. Additionally, this splitting strategy enhance the capabilities of the model user to elucidate the model's applicability domain by highlighting specific chemical categories that underwent rigorous training and testing. The distributions of temperatures and logarithmic IDAC values for the train and test sets are given in Appendix A.11.

3.5.2 Model comparison

Isothermal data subsets

Using the DECHEMA data set, a collection of isothermal subsets was constructed. First, the data points were sorted according to their temperature. Then, the data points were grouped into bins of 1 K increments. These bins were considered to be isothermal. All bins that have less than 1,000 data points were discarded, and the rest of bins are used for the present isothermal analysis. Only 9 subsets contained enough data, and they are the ones used for this extended isothermal analysis. Table 3.7 contains the number of solutes, solvents and actual observations for each isothermal subset. The size of the corresponding solute-solvent matrix is also shown together with the percentage of actual observations. Each subset's isothermal temperature was defined as the mean temperature across all corresponding data points rounded to the closest integer in degree Celsius. However, it is important to highlight that all data points in each isothermal set were kept and not averaged out further as done during the cleaning process of the DECHEMA data set. For example, if two measurements were placed in the same isothermal subset, but one was measured at, for instance, 293.15K and the other one at 293.25K, both data points were conserved in the subset. The train and test points in each isothermal

subset are determined from the splitting performed on the complete DECHEMA data set as explained in Subsection 3.5.1.

All comparisons were conducted using the natural logarithm of the IDAC ($\ln \gamma_i^\infty$). This approach not only leverages the scaling advantages of $\ln \gamma_i^\infty$ but also aligns with its intuitive presentation in chemical potential calculations (cf. Eq 2.28). Moreover, employing the natural logarithm inherently enforces the physical constraint of having positive IDACs when the re-scaling is performed.

Models' specifications

Leveraging the previously outlined isothermal subsets, a comparative analysis of phenomenological models COSMO-RS, UNIFAC-Dortmund, and MOSCED was performed. Furthermore, the proposed GNN model detailed in Section 3.3 was also evaluated, employing an ensemble of GNN models (e-GNN) trained with a 5-fold cross-validation on the training set. The final prediction of the ensemble is taken as the average prediction of the 5 models. The number of models in the ensemble, in this case 5, was chosen to balance robustness in the predictions and computational cost. As discussed in Subsection 3.3.2, the overall accuracy of the ensemble prediction tends to increase with the number of additional models included.

During the course of this research, another GNN-based architecture, referred to as SolvGNN, was proposed in the literature as a promising tool for predicting isothermal activity coefficients [125]. The SolvGNN model was also analyzed into the present extended isothermal analysis, aiming to provide insights into the distinctions and relative performance between it and the e-GNN model proposed in this work. Similarly to e-GNN, the analysis was performed using the ensemble of SolvGNN models obtained from the 5-fold cross-validation, and here denoted as *e-SolvGNN*.

Additionally, a baseline model was established by training a random forest model on the concatenated solute-solvent Morgan fingerprints, which were configured with a radius of 4 and dimensionality of 1024 bits, serving as a comparative foundation for evaluating the GNN-based models' performances. The establishment of the random forest baseline model fulfills two primary objectives. Firstly, it assesses the capability of current phenomenological models to outperform a basic data-driven methodology. Secondly, it evaluates the extent to which an advanced data-driven strategy (i.e., a GNN-based model), enhances the prediction accuracy. For this analysis, the random forest was configured with 100 estimators, utilizing the mean squared error (MSE)

Tab. 3.8.: Comparison performance of UNIFAC-Dortmund, MOSCED, COSMO-RS, the proposed e-GNN and e-SolvGNN [125] models on predicting $\ln \gamma_i^\infty$ on various isothermal subsets.

			Mean absolute error (MAE) ↓					
			Random forest	UNIFAC (Do)	COSMO-RS	MOSCED	e-GNN	e-SolvGNN
	T (K)	Feasibility (%)						
UNIFAC (Do) feasible	293.15	96.25	0.64	1.08	0.52	-	0.37	0.32
	298.15	86.73	0.62	1.23	0.54	-	0.34	0.31
	303.15	73.14	0.43	0.48	0.39	-	0.20	0.19
	313.15	77.53	0.51	0.57	0.40	-	0.24	0.22
	323.15	77.01	0.40	0.35	0.33	-	0.20	0.18
	333.15	80.90	0.42	0.43	0.36	-	0.20	0.19
	343.15	87.77	0.48	0.40	0.38	-	0.22	0.25
	353.15	85.60	0.43	0.40	0.38	-	0.21	0.20
	373.15	86.03	0.38	0.33	0.32	-	0.20	0.18
MOSCED feasible	293.15	53.44	0.60	0.47	0.38	0.31	0.33	0.26
	298.15	46.94	0.41	1.05	0.34	0.29	0.21	0.19
	303.15	21.77	0.54	0.38	0.34	0.25	0.24	0.19
	313.15	30.97	0.57	0.65	0.33	0.26	0.28	0.22
	323.15	31.23	0.45	0.37	0.30	0.17	0.18	0.15
	333.15	27.02	0.43	0.32	0.27	0.22	0.17	0.16
	343.15	22.07	0.58	0.44	0.30	0.26	0.22	0.28
	353.15	15.18	0.44	0.24	0.34	0.32	0.16	0.13
	373.15	16.91	0.60	0.22	0.28	0.36	0.21	0.22

as the training cost function. Additionally, the trees were allowed to grow until each leaf was pure.

The hyperparameter optimization for both GNN-based models (i.e., e-GNN and e-SolvGNN) models was conducted utilizing Optuna [5], incorporating 100 experimental trials alongside 10-fold cross-validation within the training set of each isothermal subset independently. Details for the hyperparameter search, along with the final hyperparameters chosen, are delineated in Appendix A.12. The same hyperparameters as in Section 3.3.1 were here optimized for the e-GNN. Similarly, for the case of e-SolvGNN, the optimized hyperparameters correspond to the ones optimized in the original work of [125]. Hyperparameters that were not optimized for both models were kept fixed in this analysis using the original reported values. The MSE was selected as the cost function for training, and Adam was selected as the optimizer to refine model weights. All computational experiments were executed on an NVIDIA Tesla P100 GPU, equipped with 16 GB of memory.

Models' performance

Table 3.8 presents a comparative evaluation of the models included in this extended isothermal study, focusing on their performance metrics measured by the mean absolute error (MAE). Unlike COSMO-RS and the data-driven models, both UNIFAC-Dortmund and MOSCED face limitations regarding their predictive capabilities. These constraints originate from the models' dependence on the availability of parameters. Consequently, these models do not have the capacity to predict all the systems covered in this extended analysis. To accommodate these limitations, Table 3.8 offers two separate comparisons: one showcases the performance across all systems in the test set that are feasible to predict by UNIFAC-Dortmund, and the other narrows the focus to those systems that are within the predictive scope of MOSCED, thereby providing a clearer view of each model's efficacy within its applicable domain. The percentage of feasible systems in the test set for UNIFAC-Dortmund and MOSCED is indicated as a feasibility percentage.

In each isothermal subset examined, the lowest MAE is highlighted in bold within Table 3.8. Notably, GNN-based methodologies outperform traditional phenomenological approaches in minimizing the MAE across all isothermal subsets. An interesting aspect of this analysis is the comparison of e-GNN's performance against MOSCED's at a temperature of 298.15 K. In this context, where the logarithmic IDACs are the focus, e-GNN demonstrates superior accuracy over MOSCED, as detailed in Table 3.8. This finding is in contrast to the observations made while using the Brouwer data set at 298.15 K (cf. Section 3.3.3), which evaluated performance based on the original scale of the IDACs, where MOSCED exhibited better results. This discrepancy highlights that upon converting predicted IDACs from the logarithmic back to the original scale, the prediction errors associated with the GNN-based model become more pronounced relative to those of the MOSCED model. This consideration becomes crucial in applications requiring the calculation of unscaled IDACs, such as when determining solvent selectivity based on the ratio of original scale IDACs.

Specifically, within the range of systems where MOSCED is applicable, it is observed that MOSCED surpasses UNIFAC-Dortmund and COSMO-RS in terms of MAE at lower temperatures. However, at high temperature the accuracy of MOSCED seems to degrade. This can be explained by the low feasibility percentage at these elevated temperatures, which affects the reliability of the comparisons.

Importantly, e-SolvGNN [125] consistently outperforms the proposed e-GNN model in the majority of isothermal subsets. The distinguishing factor between these two models lies in the explicit integration of hydrogen-bonding information. e-SolvGNN

enhances its predictive capability by constructing an interaction graph, where edges are attributed with hydrogen-bond acceptor and donor characteristics specific to the system, a feature absent in e-GNN. This distinction highlights the crucial role of including detailed intermolecular interaction data (e.g., hydrogen-bonding) in improving the accuracy of IDAC predictions. This aligns with the observations of Qin et al. [125] and with the insights from Section 3.3.3, which attributes MOSCED's strong performance to similar considerations (i.e., model parameters related to hydrogen-bonding, polarity and polarizability).

Another significant observation from the analysis involves examining the predictive accuracy of the baseline random forest model against the performances of UNIFAC-Dortmund. It can be observed that the random forest model surpasses UNIFAC-Dortmund in predictive accuracy on various instances. Specifically, at temperatures 293.15 K and 298.15 K, UNIFAC-Dortmund's predictions are notably poor. These discrepancies largely contribute to skewing the overall performance metrics of UNIFAC-Dortmund unfavorably. This highlights an important issue of UNIFAC-Dortmund, which, despite of having available binary-interaction parameters, it is still possible to get inaccurate predictions with large deviations. Predominantly, the most inaccurately predicted systems by UNIFAC-Dortmund involve interactions between solvents with pyridine groups and cyclic alkane solutes, as well as mixtures containing water as a solvent and large molecular-weight phthalate solutes. This observation suggests that for certain groups, the binary-interaction parameters are overfitted towards certain specific systems. This phenomenon was also noted in the study by Jirasek et al. [75].

When evaluating the proportion of predictions that fall within specific absolute error margins (a metric less susceptible to outlier distortions compared to the MAE) the overall findings remain consistent. Figure 3.14 elucidates this by showcasing three different absolute error thresholds (0.1, 0.2 and 0.3) for systems in the test set that can be predicted by UNIFAC-Dortmund. Once more, GNN-based methodologies distinctly outperform others across all temperatures and error thresholds. Notably, in nearly every isothermal subset and across all error benchmarks, e-SolvGNN achieves superior outcomes compared to e-GNN. This reinforces the advantage of integrating detailed molecular interaction insights into the predictive model. Furthermore, it is intriguing that despite COSMO-RS displaying lower MAE values than UNIFAC-Dortmund across the majority of temperatures (cf. Table 3.8), the scenario reverses when assessing the frequency of systems falling within the absolute error limits. This pattern suggests that COSMO-RS predictions exhibit more uniform error rates as opposed to UNIFAC-Dortmund, which shows variable predictive accuracy across different systems. Such variability in UNIFAC-Dortmund's performance might

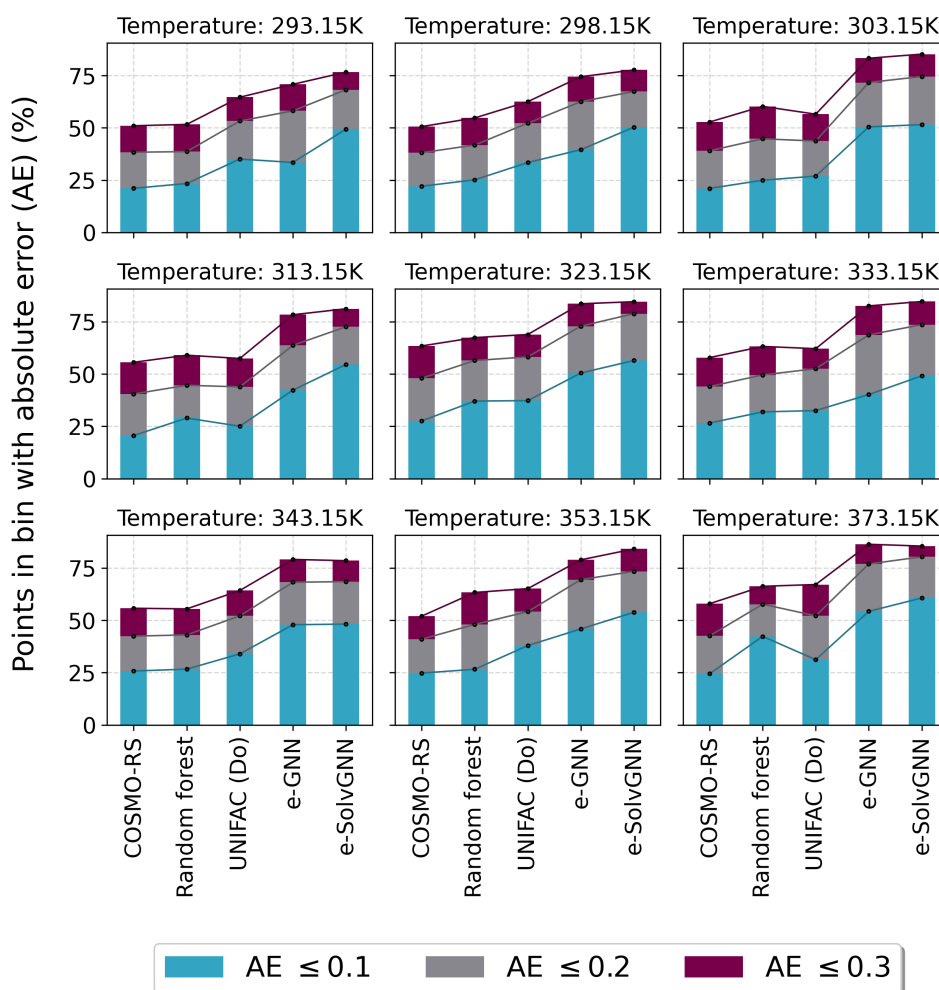


Fig. 3.14.: Percentage of systems predicted within the absolute error thresholds 0.1, 0.2 and 0.3. Values correspond to systems in the test set of the corresponding isothermal subset that are feasible to UNIFAC-Dortmund.

hint, again, at a tendency towards overfitting in several specific mixtures. Moreover, it is important to mention here, that, particularly for the UNIFAC-Dortmund model, there exists a significant overlap between the IDAC data utilized to assess its performance in this extended analysis and the IDAC data employed during the UNIFAC-Dortmund's parametrization phase [161]. Consequently, drawing conclusions about the predictive performance of the UNIFAC-Dortmund model, or indeed any group-contribution model, proves to be challenging due to the common practice of parametrizing and testing the models on the same data set.

3.6 Chapter summary

In summary, models based on GNNs emerge as potent instruments for predicting isothermal IDACs. They outperform many well-established phenomenological models, such as UNIFAC-Dortmund and COSMO-RS, in terms of prediction accuracy. However, exceptions are observed when comparing GNN models against the Abraham and MOSCED models, which are specialized models with narrower applicability domains among those evaluated. Against these models, GNNs show comparable performance in terms of accuracy, but larger applicability domain. To enhance prediction accuracy and model robustness, ensemble learning is applied to GNNs, successfully improving outcomes. Yet, the MOSCED model still exhibits competitive performance indicating that the inclusion of molecular interaction-related information as model parameters benefit the overall IDAC prediction. This is confirmed by the superior performance of e-SolvGNN in the extended isothermal analysis, which makes use of explicit hydrogen-bonding information as part of the learning framework.

The concept of a hybrid parallel GNN is introduced in this Chapter, wherein the GNN is tasked with predicting the residuals from the predictions made by phenomenological models. The final output of the hybrid model is derived by summing the phenomenological model's prediction with the correction predicted by the GNN. This methodology has been observed to enhance the accuracy of IDAC predictions at 298.15 K beyond what the individual component models achieve independently. A notable finding was that, in the context of UNIFAC-based models, employing a hybridization strategy proved less effective compared to utilizing an ensemble of GNNs designed to directly predict IDACs. Several reasons for this phenomenon are discussed in this Chapter, including the overlapping information derived from molecular structures and the propensity for significant errors in UNIFAC-based models that appear to lead the hybrid model in a detrimental direction. Despite this, the GNN-based models (either hybrid or standalone) improve the overall isothermal IDAC prediction of all phenomenological models.

It is interesting to consider, at this point, that the idea of building hybrid parallel models have been around the minds of scientists in the realm of fluid phase thermodynamics for a while. One has to simply consider the theory that has been developed around the idea of modeling a *correction* from a baseline model (e.g., ideal gas or ideal solution). Even the idea of the activity coefficient γ_i itself is built upon the idea of coupling a *correction* to a baseline. This Chapter aims to contribute in this same

direction, by showing how GNN-based models can support the pursue of modeling deviations from ideality through isothermal IDACs.

Predicting Temperature-Dependent Infinite Dilution Activity Coefficients

“ Each individual fact, taken by itself, can indeed arouse our curiosity or our astonishment, or be useful to us in its practical applications. But intellectual satisfaction we obtain only from a connection of the whole, just from its conformity with law.

— **Hermann von Helmholtz**
Physicist and physician

This Chapter advances the general objective of exploring the modeling of activity coefficients using GNNs. In the previous Chapter, the simplest scenario where the activity coefficient is estimated under constant conditions—specifically at constant temperature and at infinite dilution was explored. This Chapter continues the discussion in this direction by incorporating the actual influence of temperature variations in IDACs.

The temperature dependency of the activity coefficient is given by Eq. 2.25. This relationship is known as the Gibbs-Helmholtz equation, which relates the change of the activity coefficient due to a change in temperature to the partial molar excess enthalpy of the same species. We can write this equation also at infinite dilution conditions as follows

$$\left. \frac{\partial \ln \gamma_i^\infty}{\partial (1/T)} \right|_{P, \mathbf{x}} = \frac{\bar{h}_i^{E, \infty}}{R} \quad (4.1)$$

where, R stands for the universal gas constant and $\bar{h}_i^{E, \infty}$ is the partial molar excess enthalpy at infinite dilution.

Assuming that $\bar{h}_i^{E,\infty}$ remains constant while varying temperature, an approximation frequently validated in the literature [120], allows for the integration of the previous differential equation to yield an explicit relationship for the IDAC's temperature dependence:

$$\ln \gamma_i^\infty(T) = K_{1,i} + \frac{K_{2,i}}{T} \quad (4.2)$$

where, $K_{1,i}$ and $K_{2,i}$ are parameters that do not vary with temperature and are uniquely determined for each component i in a given solvent j . Specifically, $K_{1,i}$ represents the logarithmic IDAC at the hypothetical limit of temperature approaching infinity, while $K_{2,i}$ is defined as the ratio $\bar{h}_i^{E,\infty}/R$. The temperature-independence of $K_{1,i}$ and $K_{2,i}$ suggests that, in theory, these parameters could be derived directly from the molecular structures of the mixture components. This Chapter demonstrates such an approach by utilizing a hybrid GNN-based framework with a serial arrangement to accurately calculate these parameters based on the molecular graphs of the components. And, through this methodology, the temperature dependency of the IDAC is effectively incorporated into the model.

Furthermore, this Chapter illustrates that the proposed framework, referred to as the Gibbs-Helmholtz Graph Neural Network (GH-GNN), can be extended beyond the realm of small-sized organic molecules to model more complex chemical systems, including those with ionic liquids and polymers. These extensions aim at showing that the GH-GNN model might be broadly applied to a wide range of chemical systems of potential industrial relevance. The performance of the GH-GNN model is also compared against some of the phenomenological models introduced in the previous Chapter. The limitations of the GH-GNN model are also discussed.

4.1 Gibbs-Helmholtz Graph Neural Network (GH-GNN)

The isothermal studies presented in Chapter 3 not only highlighted the effectiveness of GNN-based models in predicting IDACs but also revealed a critical insight. Models incorporating explicit features pertaining to intermolecular interactions, especially hydrogen-bonding, such as the MOSCED and e-SolvGNN models, generally yield more accurate predictions than those lacking these considerations. However, MOSCED's major drawback is its limited applicability across the chemical space due to the constrained availability of parameters [96]. Furthermore, the GNN models discussed in Chapter 3 were confined to isothermal IDAC predictions, omitting direct

consideration of IDACs' temperature dependence. This Section aims to overcome these limitations through the introduction of the GH-GNN. The GH-GNN not only addresses the temperature dependency of IDACs through Eq. 4.2, but also integrates features related to specific intermolecular interactions, marking a significant advancement towards the overall objective of predicting activity coefficients using hybrid GNNs.

4.1.1 Data set

The data set used for training and testing the proposed GH-GNN model corresponds to the DECHEMA data set described in Section 3.5.1. The same stratified (train/test) splits are used here, which were constructed according to the chemical class allocation by Classyfire [38]. Instead of only taking the isothermal subsets, the whole DECHEMA data set is now used to leverage all available IDAC data.

Testing the partial molar excess enthalpy assumption

The premise that $\bar{h}_i^{E,\infty}$ remains constant over temperature changes is often considered a reasonable approximation for many systems, as suggested by [120]. This assumption has been previously applied to account for the temperature dependence of IDACs using matrix completion techniques [33], and was also adopted by [26] to extend their IDAC data set at 298.15 K from measurements at other temperatures. In this study, the validity of this assumption for the DECHEMA data set was examined through linear regression analysis on solute-solvent systems observed across at least three distinct temperatures, yielding a MAE on $\ln \gamma_i^\infty$ of 0.04 ± 0.099 . This outcome aligns closely with the findings of [33], who reported a MAE of 0.05 on their IDAC data set, thus indicating a potential accuracy threshold for models relying solely on Eq. 4.2 to incorporate temperature effects on $\ln \gamma_i^\infty$. However, it is also important to mention that, this assumption loses strength for systems studied over broad temperature ranges, particularly those involving highly polar components [9, 63]. Addressing this issue might involve modeling $\bar{h}_i^{E,\infty}$ directly from caloric property data sets, an area not explored in this work due to data limitations, but represents an interesting future direction for research.

Tab. 4.1.: Atomic features defining the initial feature vector of nodes in the molecular graphs constructed from the DECHEMA data set. The dimension of the corresponding one-hot encoded feature is also shown.

Feature	Description	Dimension
Atom type	[C, N, O, Cl, S, F, Br, I, Si, Sn, Pb, Ge, H, P, Hg, Te]	16
Ring	Is the atom in a ring?	1
Aromatic	Is the atom part of an aromatic system?	1
Hybridization	[s, sp, sp ² , sp ³]	4
Bonds	Number of bonds the atom is involved in. [0,1,2,3,4]	5
Charge	Atom's formal charge. [0,-1,1]	3
Attached Hs	Number of bonded hydrogen atoms. [0,1,2,3]	4
Chirality	[Unspecified, clockwise, counter-clockwise]	3

Tab. 4.2.: Bond features defining the initial feature vector of edges in the molecular graphs constructed from the DECHEMA data set. The dimension of the corresponding one-hot encoded feature is also shown.

Feature	Description	Dimension
Bond type	[Single, double, triple, aromatic]	4
Conjugated	Whether the bond is conjugated	1
Ring	Whether the bond is part of a ring	1
Stereochemistry	[None, Z, E]	1

4.1.2 Molecular graphs with global features

Molecular graphs were assembled in a manner akin to that detailed in Section 3.2, with the adaptation of atomic and bond features to fit the characteristics of compounds found in the DECHEMA data set. These features are given in Tables 4.1 and 4.2, respectively. These same atomic and bond features were used for the extended isothermal studies presented in Section 3.5, which were also constructed from the DECHEMA data set. Drawing on insights from the isothermal analysis presented in Chapter 3, this study additionally incorporated global-level features to encapsulate information potentially critical for modeling intermolecular interactions. The idea of constructing graphs with global-level features in the context of GNNs was introduced by [12]. These global-level features contain information of the complete molecule, making a graph comprehensively characterized by its node-features matrix \mathbf{A} , edge-features matrix \mathbf{B} , connectivity matrix \mathbf{C} , and a global-features vector $\mathbf{u} \in \mathbb{R}^3$.

Inspired by the parameters of the MOSCED model, the selection of global-level features, as presented in Table 4.3, includes the atomic polarizability (AP), the bond polarizability (BP) and the topological polar surface area ($TopoPSA$) of the

Tab. 4.3.: Global features defining the initial feature vector of molecular graphs constructed from the DECHEMA data set and used in the temperature-dependent IDAC studies. The dimension of the corresponding one-hot encoded feature is also shown.

Feature	Dimension
Atomic polarizability (AP)	1
Bond polarizability (BP)	1
Topological polar surface area ($TopoPSA$)	1

molecule. The AP is defined by the sum of polarizability values u^{AP} of each atom in the molecule. This cumulative polarizability is given by:

$$AP = \sum_{v \in \mathcal{V}} u_v^{AP} \quad (4.3)$$

where, v represents a non-hydrogen atom within the molecule, with u^{AP} being its specific polarizability value as reported by [68]. Similarly, BP sums the absolute differences in atomic polarizabilities across each covalent bond represented by edge e_{vw} between nodes v and w :

$$BP = \sum_{e_{vw} \in \mathcal{E}} |u_v^{AP} - u_w^{AP}| \quad (4.4)$$

where, \mathcal{E} is the set of edges in the molecular graph.

Polarizability measures how susceptible an atom or molecule is to polarization in response to an external electromagnetic field, which reflects on the strength of its dispersion forces. By incorporating these global-level features, the model is enriched with data related to dispersion interactions, akin to the induction parameter in MOSCED which aims to quantify interactions such as "dipole-induced dipole" and the ones caused by London dispersion forces in mixtures containing highly polarizable compounds. The calculation of $TopoPSA$ is based on a 2D approximation of the polar surface area, following the methodology of [121]. This global feature is akin to the polarity parameter in MOSCED, which aims at capturing mainly the "dipole-dipole" interactions. The computation of all global-level features was facilitated using the `Mordred` (version 1.2.0) computational tool [107].

4.1.3 Model architecture

In contrast to the GNN framework described in Chapter 3, several challenges must be addressed to adapt the model for broader, non-isothermal applications. A critical enhancement, as already discussed, involves explicitly incorporating temperature dependence into the learning mechanism. Furthermore, the original GNN approach transforms the solute and solvent graphs through separate GNN models, that have the same architecture but distinct parameters. Ideally, however, a GNN model for non-isothermal IDAC prediction should uniformly handle any molecular entity regardless of its (solute/solvent) role in the mixture. This unified processing is crucial having in mind a further extension of the model to scenarios involving multiple components. Additionally, and based on the performance of MOSCED and e-SolvGNN, the integration of more complex molecular interaction representations is observed to be beneficial, and hence desirable. In Section 4.1.2, the use of global-level features to construct molecular graphs that could be used to more effectively capture molecular interactions was presented. However, hydrogen-bonding information was not explicitly incorporated as part of the global-level features despite its important role on capturing the non-ideality behavior of mixtures. Instead, a specific mixture representation was used to introduce this information: the mixture graph.

Mixture graph

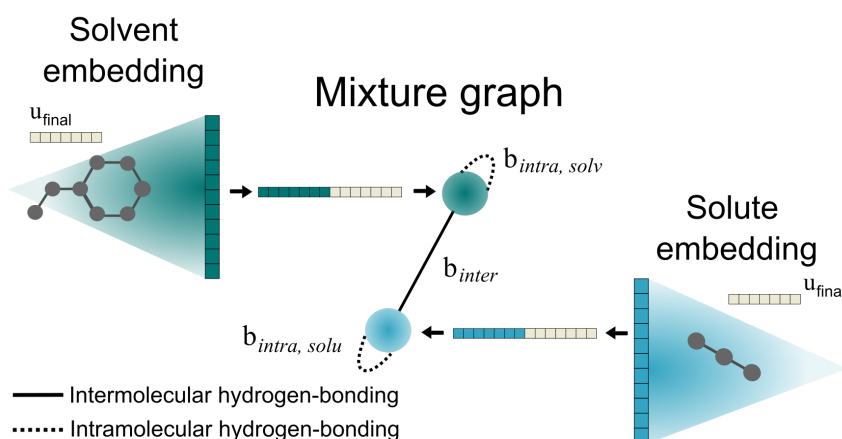


Fig. 4.1.: Schematic illustration of a two-component mixture graph.

The concept of a mixture graph was initially introduced by [125] using the term *interaction network*. The purpose of this graph is to capture a more informative representation of both intermolecular and intramolecular interactions, particularly

those involving hydrogen-bonding. Within this framework, nodes symbolize the chemical entities within a mixture, while edges denote the interactions among these entities or among themselves (in the case of self-loops). The construction of the mixture graph leverages the molecular embeddings derived from an initial GNN, now referred to as the *molecular GNN*. Therefore, nodes in the mixture graph are attributed with the corresponding molecular embeddings obtained by the molecular GNN and are concatenated to the final global-level embedding of the corresponding molecular graph, as represented in Fig. 4.1. This concept allows for a more flexible learning scheme to capture molecular interactions compared to concatenation of the molecular embeddings used in Chapter 3 (cf. Section 3.3). Following the methodology outlined by [125], hydrogen-bonding details are integrated as a single edge feature within the mixture graph. For capturing potential intermolecular hydrogen-bonding, a distinct feature, b_{inter} , is utilized. For binary mixtures, this is determined by:

$$b_{inter} = \min(N_{solv}^{HBA}, N_{solu}^{HBD}) + \min(N_{solu}^{HBA}, N_{solv}^{HBD}) \quad (4.5)$$

where, N^{HBA} represents the count of hydrogen-bond acceptors, and N^{HBD} the count of hydrogen-bond donors within a molecule, with subscripts *solv* and *solu* indicating solvent and solute, respectively. Eq. 4.5 sums the lowest count of acceptor-donor pairs across solvent and solute, effectively estimating the maximum number of feasible hydrogen-bonding sites.

For intramolecular hydrogen-bonding, this concept is similarly applied to self-loop edges in the mixture graph by employing b_{intra} , calculated as:

$$b_{intra,r} = \min(N_r^{HBA}, N_r^{HBD}) \quad (4.6)$$

with the subscript r denoting the compound in focus, either as solvent or solute. This method quantifies the intramolecular hydrogen-bonding capacity by evaluating the minimum between the available acceptor and donor sites for the specific molecule.

Serial architecture

The proposed GH-GNN model architecture consist of a hybrid GNN model constructed in a serial (or sequential) arrangement. First, for each message-passing layer, the molecular GNN model transforms the molecular graphs (enriched with global-level features) through the following 3-steps scheme:

1. Edge features update. Each vector of edge features, $\mathbf{b}_{v,w}$, within the molecular graph undergoes an update by using the embeddings from the nodes it connects (i.e., nodes v and w), its own embedding, and the global-level embedding. This process is mathematically expressed as

$$\mathbf{b}_{v,w}^{(l+1)} = \phi_b^{(l)} \left(\mathbf{a}_v^{(l)} \parallel \mathbf{a}_w^{(l)} \parallel \mathbf{b}_{v,w}^{(l)} \parallel \mathbf{u}^{(l)} \right) \quad (4.7)$$

where, \parallel symbolizes the concatenation of the vectors, while ϕ_b represents the edge updating function, which here it is a single hidden-layer neural network with the ReLU activation. Figure 4.2 graphically depicts this edge updating mechanism for the specified edge highlighted in purple color.

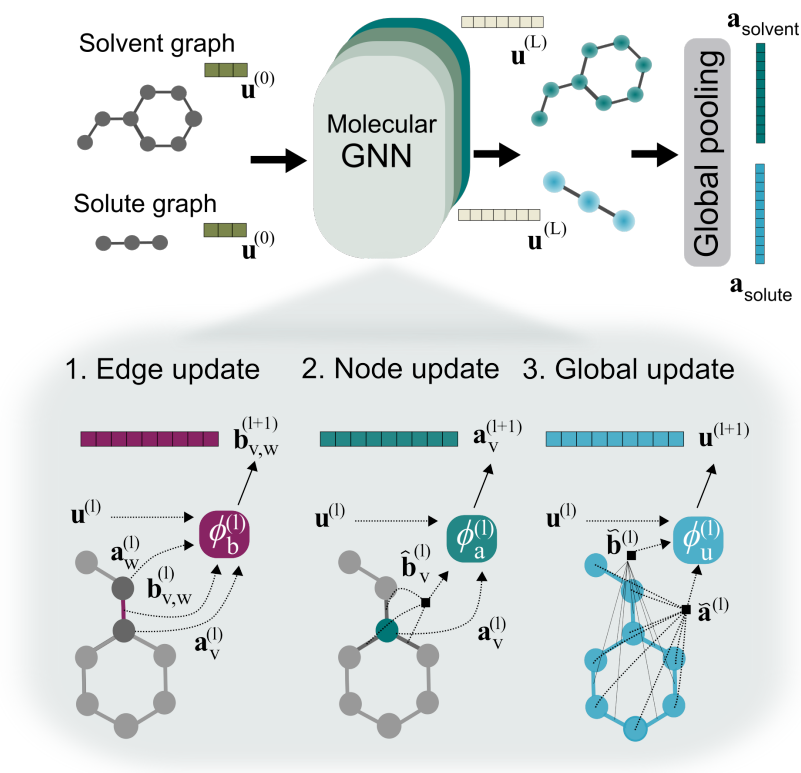


Fig. 4.2.: Schematic illustration of the molecular GNN in the Gibbs-Helmholtz Graph Neural Network.

2. Node features update. Subsequently, the embedding of each node \mathbf{a}_v is transformed by using the latest edge attributes $\mathbf{b}_{v,w}^{(l+1)}$ associated with it, its existing embedding, and global-level embedding, resulting in

$$\widehat{\mathbf{b}}_v^{(l)} = \sum_{w \in \mathcal{N}(v)} \mathbf{b}_{v,w}^{(l+1)} \quad (4.8)$$

$$\widehat{\mathbf{a}}_v^{(l+1)} = \phi_a^{(l)} \left(\mathbf{a}_v^{(l)} \parallel \widehat{\mathbf{b}}_v^{(l)} \parallel \mathbf{u}^{(l)} \right) \quad (4.9)$$

where, $\widehat{\mathbf{b}}_v$ is the vector of aggregated edge embeddings for all edges connecting node v to its neighbors $w \in \mathcal{N}(v)$. The function ϕ_a , denoting the node update mechanism, employs a single hidden-layer neural network with ReLU activation for processing. An illustration of this node update mechanism is depicted in Figure 4.2 for the green-colored node.

3. Global features update. Finally, the global embedding \mathbf{u} undergoes an update by concatenating its current state with the information from all newly updated nodes and edges within the molecular graph, as delineated by

$$\widetilde{\mathbf{a}}^{(l)} = \frac{1}{n_a} \sum_{v \in \mathcal{V}} \widehat{\mathbf{a}}_v^{(l+1)} \quad (4.10)$$

$$\widetilde{\mathbf{b}}^{(l)} = \frac{1}{n_b} \sum_{e \in \mathcal{E}} \mathbf{b}_k^{(l+1)} \quad (4.11)$$

$$\mathbf{u}^{(l+1)} = \phi_u^{(l)} \left(\mathbf{u}^{(l)} \parallel \widetilde{\mathbf{a}}^{(l)} \parallel \widetilde{\mathbf{b}}^{(l)} \right) \quad (4.12)$$

where, $\widetilde{\mathbf{a}}$ and $\widetilde{\mathbf{b}}$ symbolize the average node and edge embeddings across the molecular graph, respectively. The function ϕ_u , stands for the global update function, and is implemented through a single hidden-layer neural network equipped with the ReLU activation. Fig. 4.2 visually illustrates this global updating process for the entire molecular structure of the hypothetical solvent colored in blue. Although updating global features alter their direct physical interpretations, such as polarizability and molecular polarity, this procedure enables the GNN to assimilate significant cross-structural information across the complete graph. In all equations in the previously described 3-step updating process, the superscript (l) stands for the states at the message passing layer l , that go from the initial features (0) to the final layer (L).

For the proposed GH-GNN model, only 2 message-passing layers are used with an intermediate graph normalization, as proposed by [30], computed as

$$\mathbf{a}_v^{(l+1)} = \frac{\hat{\mathbf{a}}_v^{(l+1)} - \beta_3^{(l+1)} \odot \mathbb{E}[\hat{\mathbf{a}}_v^{(l+1)}]}{\sqrt{\text{Var}[\hat{\mathbf{a}}_v^{(l+1)} - \beta_3^{(l+1)} \odot \mathbb{E}[\hat{\mathbf{a}}_v^{(l+1)}]]} + \epsilon} \odot \beta_1^{(l+1)} + \beta_2^{(l+1)} \quad (4.13)$$

where, $\mathbb{E}[\cdot]$ and $\text{Var}[\cdot]$ represent the expectation and variance operators, ϵ is a small number added for numerical stability, \odot denotes element-wise multiplication and $\beta_1^{(l+1)}$ and $\beta_2^{(l+1)}$ are learnable vector parameters that scale and shift the normalized node embeddings.

Compared to the batch normalization (Eq. 3.3) used in Chapter 3, the graph normalization used here introduces an extra vector of learnable parameters $\beta_3^{(l+1)}$ that weights the amount of information that needs to be preserved from the mean value of the node embeddings. This parameter allows for an adaptive scaling of features, which can help in preserving node-specific information that might be important for the downstream task. This information could be lost in the case of a standard batch normalization process. Moreover, the expectation and variance operators in the batch normalization from Chapter 3 are calculated from the training mini-batch, while, here, they are computed for each individual graph. It has been observed that this type of graph normalization decreases the influence of the batch noise and increases the expressiveness of the graph representation learning process [30].

After L message-passing layers, a global pooling operation is conducted (for each molecular graph) that averages the final node embeddings to compute a molecular graph embedding

$$\mathbf{a}_g = \frac{1}{n_a} \sum_{v \in \mathcal{V}} \mathbf{a}_v^{(L)} \quad (4.14)$$

After this, the mixture graph is constructed as explained in Section 4.1.3, where nodes are attributed by the concatenation of the corresponding graph embedding \mathbf{a}_g and the final global-level embedding $\mathbf{u}^{(L)}$

$$\mathbf{a}_{mg}^{(0)} = \mathbf{a}_g \parallel \mathbf{u}^{(L)} \quad (4.15)$$

where, the subscript mg indicates that the node embedding corresponds to the node in the mixture graph. And edges are attributed with the features obtained from Eq. 4.5 or 4.6, depending on whether they interconnect the nodes or connect the node with itself, respectively.

Once the mixture graph is constructed, the node embeddings are mapped to the dimensions of the second GNN, referred to as the *mixture GNN*, using a linear transformation followed by a ReLU activation as follows

$$\mathbf{a}_{mg}^{(l)} = \text{ReLU} \left(\mathbf{W}^{(0)} \cdot \mathbf{a}_{mg}^{(0)} + \mathbf{q}^{(0)} \right) \quad (4.16)$$

where, $\mathbf{W}^{(0)}$ and $\mathbf{q}^{(0)}$ are learned as part of the training process.

Then, the mixture GNN implements a message passing scheme similar to the one proposed in [52] and to the one used in Eq. 3.2, which can be written as

$$\mathbf{a}_{mg}^{(l+1)} = \text{GRU} \left(\mathbf{W}^{(l)} \cdot \mathbf{a}_{mg}^{(l)} + \sum_{w \in \mathcal{N}(v)} \left(\phi_E^{(l)}(\mathbf{b}_{vw}) \cdot \mathbf{a}_w^{(l)} + \mathbf{q}^{(l)} \right) \right) \quad (4.17)$$

where, $\phi_E^{(l)}$ is a single hidden-layer neural network with the ReLU activation function, which processes the edge feature vector \mathbf{b}_{vw} between nodes v and $w \in \mathcal{N}(v)$. Notice, that in the case of the mixture graph, which contains self-loop connections, the node in question is also part of the sets of neighboring nodes, i.e., $v \in \mathcal{N}(v)$. The GRU stands for the gated recurrent unit and acts here as the update function.

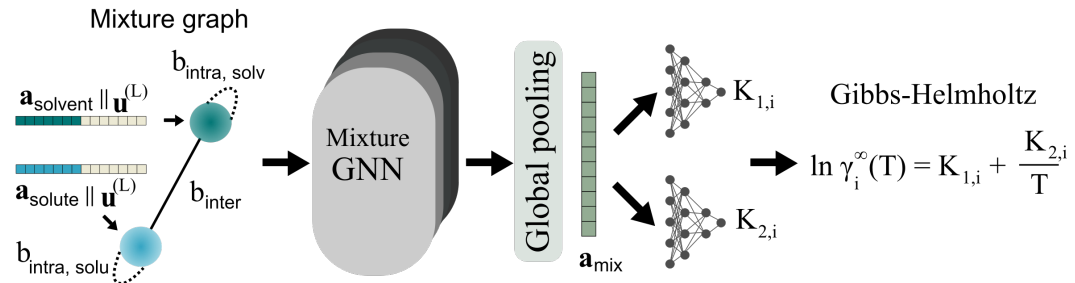


Fig. 4.3.: Schematic illustration of the mixture GNN in the Gibbs-Helmholtz Graph Neural Network.

Finally, the updated node embeddings of the mixture graph are passed through a global pooling operation according to

$$\mathbf{a}_{mix} = \mathbf{a}_{mg,solv}^{(L)} \parallel \mathbf{a}_{mg,solu}^{(L)} \quad (4.18)$$

where, \parallel denotes concatenation and the subscripts *solv* and *solu* refer to the solvent and solute nodes, respectively. This mixture embedding \mathbf{a}_{mix} is used to regress the

parameters of Eq. 4.2 through two separate MLP models consisting of 2-hidden layers with the ReLU activation function. This provides the physical framework to introduce the temperature dependency into the IDAC modeling. Therefore, the proposed GH-GNN model can be broadly understood as the sequential connection of the molecular GNN (depicted in Fig. 4.2), the mixture GNN and the Gibbs-Helmholtz derived expression given by Eq. 4.2 (both depicted in Fig. 4.3).

4.1.4 Multi-task pre-training

To fully exploit the use of Eq. 4.2 for introducing the temperature dependency, the training of GH-GNN incorporates a two-step process leveraging transfer learning. Initially, the model undergoes pre-training in a multi-task framework aimed at predicting the parameters $K_{1,i}$ and $K_{2,i}$ from Eq. 4.2. This phase employs the sum of mean squared errors as its loss metric, and it uses the pre-calculated values for these parameters through linear regression (previously calculated to validate the assumption of constant $\bar{h}_i^{E,\infty}$ over temperature). This pre-training step mirrors the methodology of the matrix completion method (MCM) outlined by [33]. The same normalization of $K_{1,i}$ and $K_{2,i}$ procedure as described by [33] was here used. Subsequently, the model is fine-tuned on the complete training set to predict $\ln \gamma_i^\infty$, utilizing the parameters refined during pre-training.

Distinct from the MCM, which limits its training scope to systems observed across a minimum of three temperatures [33], GH-GNN is able to use the entirety of the available IDAC data as part of the fine-tuning phase. This enables the model to simultaneously learn the variation of IDACs over temperature for different systems, while also capturing broader solute-solvent interactions from the data. For example, for systems observed at only one temperature, where MCM would exclude them, the GH-GNN model incorporates these data points to broaden its learning across a broader array of molecular compounds. It then infers the temperature dependency from the collective information of other systems measured under various temperatures. This approach ensures the maximum utilization of all experimental data, a highly desirable characteristic given its scarcity.

4.1.5 Model performance

To evaluate the performance of the proposed GH-GNN model, a series of comparisons against the phenomenological model UNIFAC-Dortmund and against other GNN models were conducted. Moreover, the proposed GH-GNN model was tested

for a diverse set of prediction tasks, including the inter-/extrapolation to other temperatures, and the inter-/extrapolation to other chemical compounds. These comparisons are detailed in the following Subsections.

Performance on DECHEMA data set

The performance of the GH-GNN model on the test set was compared to 3 other GNN-based models, and to UNIFAC-Dortmund. Each of these GNN-based models is described below highlighting the differences compared to the GH-GNN model and the main reasons that inspired the comparisons for this analysis.

GNNCat. This model is analogous to GH-GNN by following the same architecture. The key distinction lies in the treatment of the mixture embedding, \mathbf{a}_{mix} . After computing \mathbf{a}_{mix} , GNNCat augments it by concatenating the normalized temperature, T , of the mixture, resulting in a vector defined as:

$$\mathbf{a}_{mix}^{Cat} = \mathbf{a}_{mix} \parallel T \quad (4.19)$$

In contrast to employing 2 MLPs, GNNCat utilizes a single MLP that receives \mathbf{a}_{mix}^{Cat} as its input for predicting $\ln \gamma_i^\infty$. This design choice facilitates a direct comparison, aimed at evaluating the efficacy of integrating the IDAC temperature dependency into the calculation using Eq. 4.2 versus incorporating the temperature as an additional input parameter to the MLP. The latter approach has been used by [132] for the prediction of IDACs in systems containing ionic liquids. To ensure an equitable comparison between the GH-GNN and GNNCat models, the number of trainable parameters were set the same for both models. This was achieved by doubling the number of neurons in the first layer of the single MLP of GNNCat, aligning the models' learning capacities for a fair evaluative basis.

GH-SolvGNN. This model is an adaptation of the SolvGNN model, initially introduced by [125] and used in the additional isothermal studies described in Section 3.5. This adaptation extends the original SolvGNN, which was designed to process only isothermal conditions, to also accommodate temperature variations using Eq. 4.2. Similar to the GH-GNN model, GH-SolvGNN employs two MLPs for predicting the parameters defined in Eq. 4.2. One of the limitations of the original SolvGNN architecture was its inability to integrate global-level features within molecular graphs. To address this, GH-SolvGNN enhances the node embeddings of the mixture graph by

Tab. 4.4.: Performance comparison of GNN-based models and UNIFAC-Dortmund for the prediction of temperature-dependent IDACs.

Entire test data set				
Model	MAE ↓	$AE \leq 0.1 \uparrow$	$AE \leq 0.2 \uparrow$	$AE \leq 0.3 \uparrow$
GNNCat	0.13	64.97%	84.44%	91.47%
GH-GNN (w/o pre-training)	0.15	58.7%	81.4%	89.66%
SolvGNNCat	0.13	65.55%	84.15%	90.85%
GH-SolvGNN (w/o pre-training)	0.14	63.52%	82.72%	89.72%
GH-GNN	0.12	73.68%	87.13%	92.22%
UNIFAC (Do) feasible systems in the test data set				
GNNCat	0.13	63.91%	84.05%	91.29%
GH-GNN (w/o pre-training)	0.15	57.84%	80.71%	89.40%
SolvGNNCat	0.14	65.02%	83.66%	90.47%
GH-SolvGNN (w/o pre-training)	0.15	62.97%	82.21%	89.27%
GH-GNN	0.12	72.37%	86.20%	91.75%
UNIFAC (Do)	0.60	33.10%	51.76%	64.32%
Other models in other data sets				
MCM[33]	-	-	-	(76.6%)
SPT[166]	(0.11)	-	-	(94%)

concatenating the global-level features described in Table 4.3. The objective of contrasting GH-GNN with GH-SolvGNN is to determine the extent to which the choice of GNN architecture influences the accuracy of predicting temperature-dependent IDACs.

SolvGNNCat. This model is designed analogously to GNNCat, but with respect to the GH-SolvGNN framework. In the same way, this adaptation simplifies the predictive mechanism of GH-SolvGNN by utilizing a single MLP instead of two. This MLP processes the mixture embeddings concatenated with the normalized temperature, as outlined in Equation 4.19, to predict $\ln \gamma_i^\infty$ directly. The purpose of including this model in the comparison is to broaden the observations into how different strategies for incorporating temperature information affects the model's performance.

The hyperparameters of the GNN-based models were optimized using `Optuna` [5] over 100 trials using 10-fold cross-validation on the training set. The final hyperparameters, the ranges explored and further details on the hyperparameter optimization process are available in Appendix A.13. The MSE was used as the loss function and AdamW [101] as the optimizer. All the numerical studies were performed on a single NVIDIA Tesla P100 GPU (16 GB).

Table 4.4 showcases the comparative analysis of the GNN-based models and the UNIFAC-Dortmund method. The comparison is based on the MAE and the proportion

of predictions within specified absolute error margins (0.1, 0.2, and 0.3). The table distinguishes between models trained directly to predict $\ln \gamma_i^\infty$, marked as "(w/o pre-training)" to denote the absence of the preliminary multi-task training phase, and models trained using the 2-step transfer learning process described in Section 4.1.4. The performance is given for the complete test set, and for the subset of systems in the test set that UNIFAC-Dortmund is capable of predicting, which constitutes approximately 84% of the original test data set. Additionally, the table includes the performances of other state-of-the-art data-driven approaches, namely the matrix completion method (MCM) [33] and the SMILES-to-Property-Transformer (SPT) [166], which also aim to predict $\ln \gamma_i^\infty$ across different temperatures. It is important to note, as further detailed in Section 3.5.1, that both MCM and SPT were evaluated on distinct, and considerably smaller, data sets compared to the one considered here. Therefore, the values are provided for an overview purpose rather than for a direct comparison. The best value for each comparison is highlighted in bold.

The performance comparison reveals that models trained with temperature concatenation (i.e., GNNCat, SolvGNNCat) outperform their counterparts developed without the pre-training step. This discrepancy suggests that models based on Eq. 4.2 struggle to attain optimal parameter sets during training as opposed to the more flexible approach of temperature concatenation. However, with the inclusion of the multi-task pre-training step, the GH-GNN model surpasses all other models across all metrics. Considering the experimental uncertainty estimation of the logarithmic IDAC being between 0.1 and 0.2 [33], the GH-GNN model accurately predicts over 87% of systems within this range. If a similar multi-task pre-training step is implemented for GH-SolvGNN the accuracy and robustness would likely increase similarly to GH-GNN.

As discussed in Section 3.5.2, the relatively high MAE exhibited by UNIFAC-Dortmund is attributed to its severe mispredictions for systems containing pyridines, quinolines, and water. Despite this, UNIFAC-Dortmund's performance remains inferior to GNN-based models in terms of accurately predicting systems within the error thresholds, a more robust metric against outliers compared to MAE.

To assess the impact of the polarity and polarizability information included in the modeling framework, the performance of a GH-GNN model that uses randomly generated global-level descriptors during pre-training and fine-tuning was also measured. The GH-GNN model with random global features yielded a test MAE of 0.15, with 89.43% of points having an absolute error below 0.3. This suggests that besides hydrogen-bonding information, including polarizability and polarity increases the accuracy of IDAC predictions. This might be related to the effective

capturing of weaker (but still relevant) molecular interactions, such as "dipole-induced dipole" and "induced dipole-induced dipole" interactions. Notably, the GH-GNN model trained with random global features performed worse than GNNCat, underscoring the importance of including such descriptors for prediction quality.

Although not directly comparable, the SPT model's performance [166] appears to be on the same order of that of GH-GNN. However, the GH-GNN model achieves its performance across a broader and more diverse chemical space, as detailed in Table 3.6. Moreover, while SPT relies on computationally intensive pre-training procedures involving millions of COSMO-RS-simulated data points followed by fine-tuning using experimental data, the proposed GH-GNN model is trained directly on experimental data. Integrating a similar pre-training step utilizing COSMO-RS data could potentially further enhance the GH-GNN's accuracy. Additionally, the pre-trained GH-GNN model's performance without the fine-tuning phase (i.e., trained solely for predicting the $K_{1,i}$ and $K_{2,i}$ parameters of Eq. 4.2) results in 78.86% of systems predicted with an absolute error (AE) of ≤ 0.3 , a performance level similar to that reported by the MCM model (cf. Table 4.4) in their considered data set.

Performance on predicting IDACs at other temperatures

In order to assess the ability of the proposed GH-GNN model to generalize to temperatures beyond those encountered during training, three different scenarios were considered. First, the interpolation performance was examined. For this, all solute-solvent systems in the test set that have a temperature T_* and that are also present in the training set at temperatures T_1 and T_2 , such that $T_1 \leq T_* \leq T_2$ were included in the analysis. Second, the extrapolation to high temperatures was examined. Here, systems in the test set with temperature T_{high} that are also contained in the training set, but only at temperatures lower than T_{high} are included. Third, the extrapolation to low temperatures was studied. In this case, systems in the test set with temperature T_{low} that are also contained in the training set, but at temperatures higher than T_{low} are analyzed. Table 4.5 show the MAE and the percentage of points predicted below an absolute error of 0.3 achieved by the GH-GNN model in the test set in each of these scenarios. For comparison, the performance of the GNNCat model is also shown. The best value is highlighted in bold. Moreover, the number of included data points and the proportion to the total test set are shown for each scenario.

It can be seen that the GH-GNN model generally outperforms GNNCat in both interpolating and extrapolating tasks to various temperatures. Notably, both models

Tab. 4.5.: Performance of the GH-GNN and GNNCat models for predicting IDACs while interpolating among temperature values, extrapolating to lower temperatures and extrapolating to higher temperatures.

Model	MAE ↓	AE < 0.3 ↑	# data points	% of test set
Interpolation				
GNNCat	0.11	93.12	3025	36.41%
GH-GNN	0.10	92.99	3025	
Extrapolation to T_{high}				
GNNCat	0.11	94.06	1684	20.27%
GH-GNN	0.10	94.54	1684	
Extrapolation to T_{low}				
GNNCat	0.14	91.30	1954	23.52%
GH-GNN	0.11	93.54	1954	

excel in predicting IDACs of solute-solvent systems that were observed as part of the training but at different temperatures. Nevertheless, even when both models succeed on predicting IDACs at different temperatures, the fact that the GH-GNN model uses Eq. 4.2 for introducing the temperature allows for a faster computation of IDACs of a system at various temperatures compared to the approach of GNNCat. This advantage might be small in scenarios where direct IDAC predictions are needed, but it is significant in scenarios where iterating over temperatures is needed (e.g., in computations of isobaric vapor-liquid equilibria).

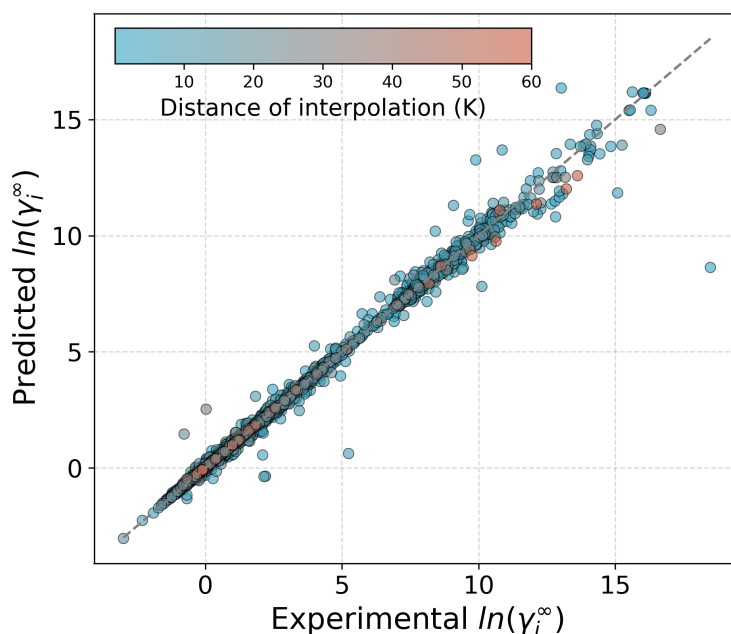


Fig. 4.4.: Parity plot between the experimental and the predicted IDAC values by the GH-GNN while interpolating to other temperatures.

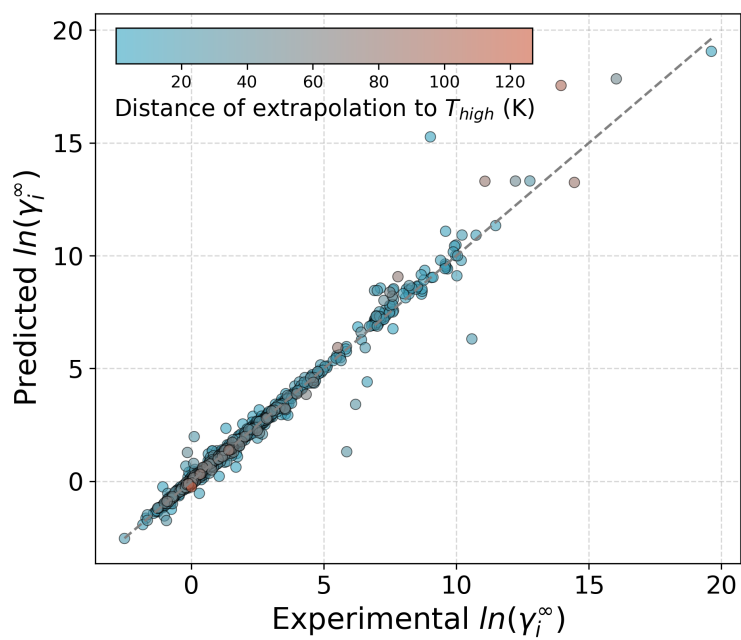


Fig. 4.5.: Parity plot between the experimental and the predicted IDAC values by the GH-GNN while extrapolating to temperature T_{high} .

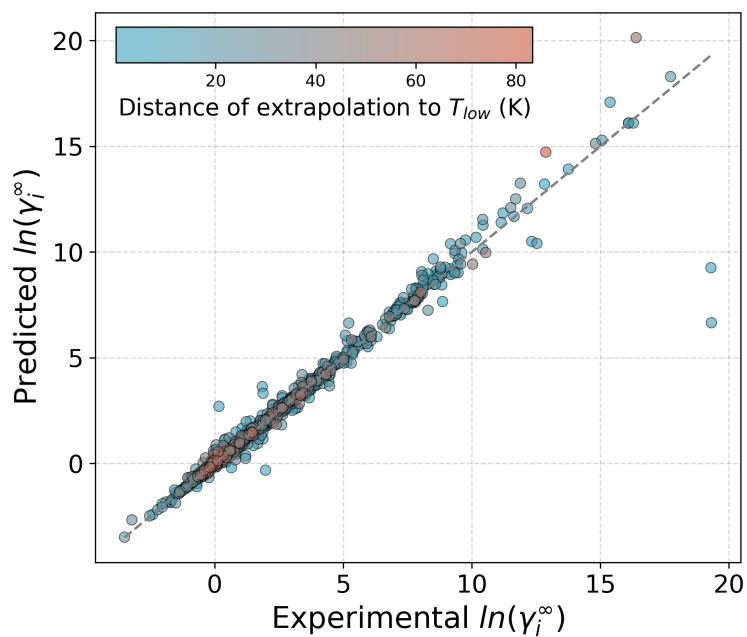


Fig. 4.6.: Parity plot between the experimental and the predicted IDAC values by the GH-GNN while extrapolating to temperature T_{low} .

For a clearer illustration, Figures 4.4, 4.5 and 4.6 show the parity comparison for each of the scenarios described above. The color map indicates the minimum Euclidean distance between the temperature of the system in the test set and that in the training set. From the parity plots of the three cases it is clear that the accuracy of the predictions is not directly correlated to the distance of temperature inter-/extrapolation. This is confirmed by the multiple mispredictions that have a relatively small temperature distance to the training data. Instead, the accuracy of the predictions is more related with the amount of data in the specific regime, and the type of chemicals being involved. Since data for highly non-ideal systems (i.e., $\ln \gamma_i^\infty \geq 10$) is very scarce (cf. Appendix A.11) predictions of such systems tend to be deficient. Moreover, the mispredictions observed at $\ln \gamma_i^\infty < 10$, involve systems with a protic solvents, such as water and methanol. Therefore, studying the performance of the GH-GNN model, or any GNN-based model tailored for IDAC predictions, for the prediction of different chemical species appears to be more important to accurately assess the capacities and limitations of the model.

Performance on predicting IDACs of different binary-systems

In order to test the performance of the GH-GNN model on predicting IDACs of systems that have not being explicitly seen during training, two different tasks were designed. First, the interpolation among binary-systems was tested. This refers to predicting systems where both the solute and the solvent species are contained in the training set, but not in the precise combination as in the test set. This task is akin to the matrix completion problem [33], in which the predictions are strictly limited to species that have been seen during training. Second, the performance of the GH-GNN model is tested when predicting systems where either the solute or the solvent have not been seen during training at all. This could be thought as predicting IDACs outside of the rows (or columns) of the solute-solvent matrix defined during training, and it is here referred to as extrapolating to different binary systems.

Figure 4.7 shows the parity comparison of the GH-GNN predictions when tasked with interpolating among binary-systems. The number of systems in the test set that fall into this prediction task are 1,568 (18.87% of the test set). Remarkably, as pointed out in Table 4.6, the GH-GNN model achieves a MAE of 0.13, with 88.84% of the systems exhibiting an absolute error below 0.3. By comparison, the MCM, operating on a smaller solute-solvent matrix, achieves a lower performance, predicting only 76.6% of the systems below an absolute error of 0.3 [33] (cf. Table 3.6). Moreover, Table 4.6 shows the performance of the UNIFAC-Dortmund model on all feasible systems and excluding its worst 9 predictions (indicated by “w/o”). It is evident that,

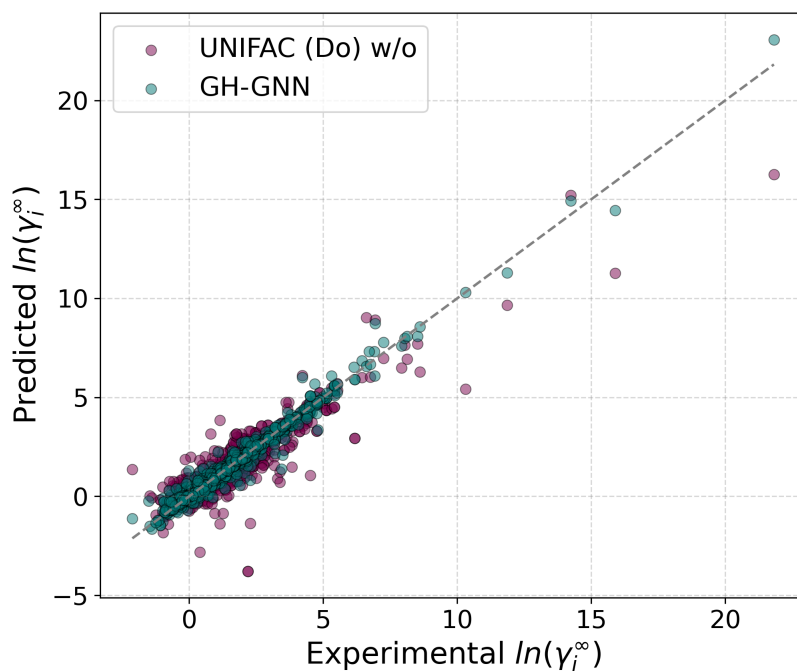


Fig. 4.7.: Parity plot between the experimental and the predicted IDAC values by the GH-GNN and the UNIFAC-Dortmund models while interpolating among binary-systems. The results for UNIFAC-Dortmund are shown for all feasible systems in the test set excluding the worst 9 predictions.

when considering the task of interpolating to other binary-systems, the GH-GNN model outperforms UNIFAC-Dortmund in terms of prediction accuracy (measured by the MAE), prediction robustness (measured by the percentage of predicted systems with an error below 0.3) and number of feasible systems that is able to predict. This is true even when the worst outliers of UNIFAC-Dortmund are excluded from the analysis.

Tab. 4.6.: Performance of the GH-GNN and UNIFAC-Dortmund models for predicting IDACs while interpolating among binary-systems. Results excluding the worst 9 predictions are indicated by “w/o”.

Model	MAE ↓	AE ≤ 0.3 ↑	# feasible data points	% feasible systems
UNIFAC (Do)	0.79	64.41	1270	80.99%
UNIFAC (Do) (w/o)	0.36	64.87	1261	80.42%
GH-GNN	0.13	88.84	1568	100%

Table 4.7 show the results of the GH-GNN and the UNIFAC-Dortmund models when extrapolating to other binary-systems. As can be seen the performance of the predictions worsen considerably. However, it is important to note that the number of systems that were available within the test set to test the extrapolation to other systems is very limited (i.e., only 77 systems). Therefore, the generalization of this

comparison has to be considered with care. Despite this, for the systems analyzed here, the GH-GNN model outperforms UNIFAC-Dortmund in accuracy, robustness and feasibility, even when removing the worst 9 predictions of UNIFAC-Dortmund. It is important to highlight that approximately half of the systems analyzed in this extrapolation study involve water. Predicting water-containing systems has been particularly challenging due to the strong interactions that water molecules engage in, which results in a broad range of possible IDAC values [85, 166]. If one removes all water-containing systems, the MAE achieved by the GH-GNN model comes down to 0.25. In contrast, the UNIFAC-Dortmund predicts such systems with a MAE of 0.44.

Tab. 4.7.: Performance of the GH-GNN and UNIFAC-Dortmund models for predicting IDACs while extrapolating to other solutes or solvents. Results excluding the worst 9 predictions are indicated by “w/o”.

Model	MAE ↓	AE ≤ 0.3 ↑	# feasible data points	% feasible systems
UNIFAC (Do)	2.41	23.61	72	93.51%
UNIFAC (Do) (w/o)	1.26	26.98	63	81.82%
GH-GNN	1.11	46.75	77	100%

Given the limited number of systems available for testing extrapolation, a new data set was constructed specifically for testing the extrapolation capabilities of the GH-GNN model to other mixtures. This new data set, here referred to as the *external data set* was constructed from the experimental IDAC data originally gathered by [26] and further cleaned by [166]. For the purposes of this analysis only organic systems were included. Also, any system containing molecules with atomic or bond features that were not feasible by the features established in Tables 4.1 and 4.2 were discarded. Additionally, only systems where either the solute or the solvent was not present in the DECHEMA training set were kept, and repeated measurements were averaged to obtain a single value per system at a specific temperature. The resulting external data set consists of 2,058 data points.

The GH-GNN model achieved a MAE of 0.43 on the external data set, with 56.90% of the data being predicted below an absolute error of 0.3. Figure 4.8 shows the parity performance. Additionally, to illustrate how the accuracy of the predictions is correlated with the rarity of the extrapolated system, the number of systems in the training set that have the same solute-solvent chemical classes is shown as the color map. It can be qualitatively observed that systems with popular classes in training are better predicted compared to those containing chemical classes barely seen during training.

The GH-GNN model attained a MAE of 0.43 on the external data set, with 56.90% of the predictions having an absolute error of 0.3 or lower. The parity performance

is depicted in Fig. 4.8. Moreover, to illustrate the correlation between prediction accuracy and the rarity of extrapolated systems, the color map represents the number of systems in the DECHEMA training set with the same solute-solvent chemical classes. It can be observed qualitatively from the visualization that systems belonging to commonly occurring classes in the training data tend to be predicted more accurately compared to those with chemical classes that are less prevalent in the training set.

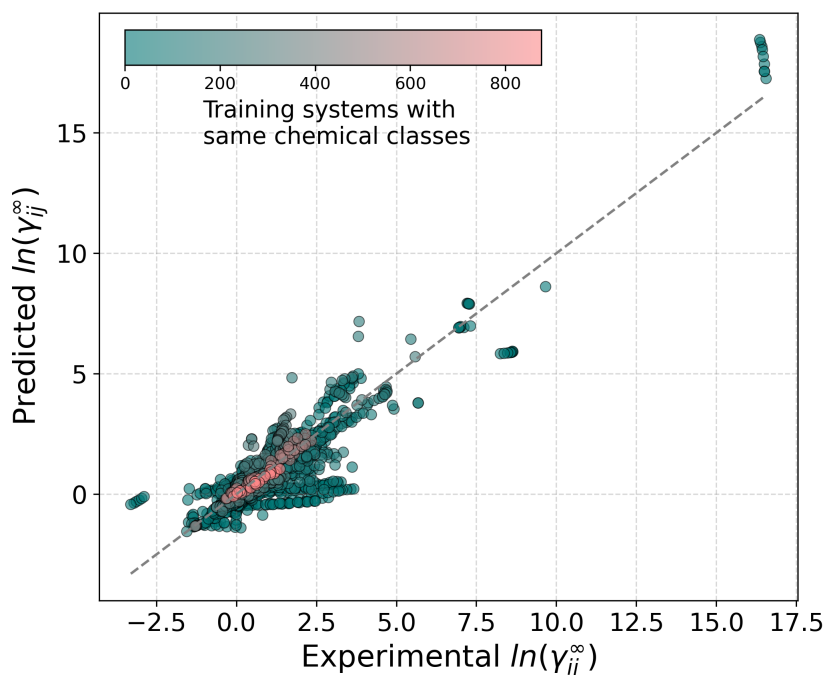


Fig. 4.8.: Parity plot between the experimental and the predicted IDAC values by the GH-GNN model on the external data set.

In order to have a more quantitative measure of how correlated the rarity of the extrapolated systems is with respect to the prediction accuracy, an *extrapolation distance* metric δ_s was computed for each extrapolated species s in the external data set. This metric is defined as

$$\delta_s = \frac{1}{|\mathcal{T}|} \sum \{d_{s,k} \mid k \in \mathcal{T}\} \quad (4.20)$$

$$d_{s,m} = 1 - \text{Jaccard}(FP_s, FP_m) \quad (4.21)$$

$$\text{Jaccard}(FP_s, FP_m) = \frac{|FP_s \cap FP_m|}{|FP_s \cup FP_m|}, \quad \forall m \in \mathcal{M} \quad (4.22)$$

where, \mathcal{T} is the set of 10 shortest Jaccard distances $d_{s,m}$ between the fingerprint FP_s of the extrapolated species s and the fingerprint FP_m of each species m in the DECHEMA training set \mathcal{M} . The Jaccard similarity measures the ratio of the size of the intersection of two sets to the size of their union, representing the proportion of elements that differ between them. The fingerprints were computed using the RDKit fingerprint [127] (version 2021.03.1).

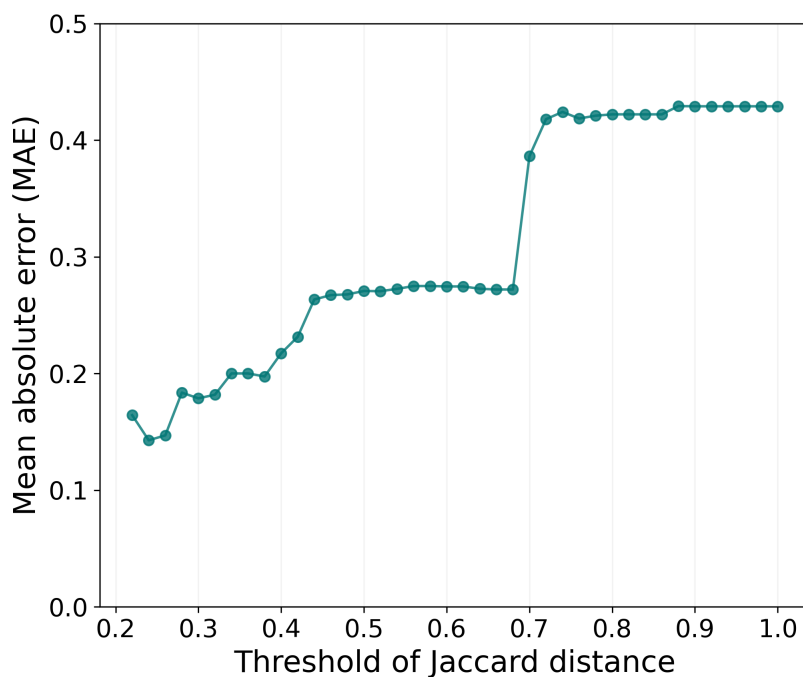


Fig. 4.9.: Progression of the mean absolute error (MAE) achieved by the GH-GNN model on systems in the external data set that fall into different Jaccard distance thresholds.

Figure 4.9 illustrates the MAE progression achieved by the GH-GNN model across systems in the external data set where δ_s is equal to or less than the specified threshold. As evident, there exist a trend of increasing MAE with distance, which aligns with the common understanding that the performance of data-driven models deteriorates as they extrapolate further from training distribution. The pronounced drops in MAE progression as the distance threshold tightens can be attributed to the specific chemical diversity covered in the external data set compared to the DECHEMA training data set. Interestingly, as the Jaccard distance threshold approaches values below 0.3, some oscillation occurs. This happens particularly around MAE values between 0.1 and 0.2, roughly corresponding to the experimental uncertainty estimation of [33]. Despite these limitations in interpretability, the accessibility of chemical class information and the observed correlation between prediction accuracy and the proposed Jaccard distance metric, provide a valuable

framework for model users to understand its applicability domain and gain insights into its expected performance.

4.2 Extending the GH-GNN model to ionic liquids

In all previous discussions, the GH-GNN model was tested exclusively on organic molecules. All systems containing ionic liquids (ILs) were excluded. However, ionic liquids have encountered great interest in several chemical engineering applications due to their particular thermophysical properties (e.g., very low vapor pressure, high thermal stability), specially in separation processes [18]. In this Section, the extension of the proposed GH-GNN model for predicting systems with ionic liquids as solvents is presented. A GNN-based model and a MCM-based model were reported in the literature recently [132] for predicting exactly this type of systems. Therefore, the comparison of the extended GH-GNN model and those available from the literature is also included. Moreover, the data-driven models are compared to the phenomenological model UNIFAC-IL [32].

4.2.1 Data set

The data set utilized for the extension of the GH-GNN model corresponds to the data available at the ILThermo (v2.0) database [80]. This data set has previously been employed for developing the GNN and MCM models by [132], and for the parametrization of the most extensive version of UNIFAC-IL [32]. It comprises 215 ionic liquids (ILs), consisting of 96 cations and 38 anions, alongside 112 solutes, yielding a total of 41,553 experimental IDAC values. The temperatures covered in this data set go from 288.15 to 448.15 K, with a median temperature of 338.15 K. For consistency, the same training and testing split utilized by [132] to assess interpolation performance among binary systems was here utilized.

4.2.2 Extension strategies

Three distinct extension strategies were explored for applying the proposed GH-GNN framework to ILs. The first strategy involved training a GH-GNN model directly on IL data. The second strategy leveraged the GH-GNN model developed for organic mixtures as a pre-trained model, which is later fine-tuned using the IL data. The third strategy entailed re-training the model using the combined data from both

organic systems and ILs simultaneously. The objective behind comparing these strategies was to discern whether a GNN-based model tailored to specific system types (in this instance, organic versus ionic) outperformed a more generalized implementation capable of predicting both systems simultaneously. Consistent with the original GH-GNN model, identical model architecture, hyperparameters, and training specifications were maintained across all extension strategies, as delineated in Sections 4.1.3 and 4.1.4. Only the multi-task pre-training step was excluded for the different extension strategies to ILs.

Tab. 4.8.: Performance of different models for predicting IDACs of systems containing ionic liquids (ILs).

Model	Organic systems		IL systems	
	MAE ↓	AE ≤ 0.3 ↑	MAE ↓	AE ≤ 0.3 ↑
GH-GNN	0.12	91.75%	1.21	16.80%
GH-GNN (direct IL)	2.06	10.24%	0.09	94.31%
GH-GNN (pre-trained organic)	1.67	10.94%	0.09	94.12%
GH-GNN (simultaneous)	0.12	92.38%	0.09	93.98%
GNN [132]	-	-	0.09	93.84%
MCM [132]	-	-	0.09	94.14%
UNIFAC-Dortmund	0.60	64.32%	-	-
UNIFAC-IL [32]	-	-	0.49	50.53%

Table 4.8 shows the comparison performance of the original GH-GNN model (including the multi-task pre-training step) and the models developed with the three extension strategies explained before. Additionally, the performance of the GNN and MCM models developed by [132] are shown. In order to compare the performance of the developed models to phenomenological approaches, the UNIFAC-Dortmund and UNIFAC-IL [32] models are also included. The comparisons are shown for the test set of organic systems (i.e., DECHEMA test set) and for IL systems.

Several interesting insights can be observed from this extension to ILs. First, it is evident, that the original GH-GNN model, even though performing excellent for predicting organic systems, it cannot accurately extrapolate to IL systems. This is expected, as ionic liquids participate in stronger electrostatic interactions compared to organic molecules, and they tend to be larger in size than the organic counterparts. A second insightful observation is that an analogous behavior is observed for the GH-GNN model directly trained on the IL data. The model performs the best for predicting IL systems, but it poorly extrapolates to organic systems. Therefore, if a general data-driven model is to be build for predicting IDACs of organic and IL systems simultaneously, both types of systems must be included into the training data set.

In the second strategy, transfer learning was used in which the original GH-GNN model was used as the pre-trained model, which was later fine-tuned with IL data. This model maintains a comparable performance when predicting IL systems compared to the first strategy. Moreover, it also improves the performance on predicting organic systems compared to the first extension strategy. Nevertheless, the performance on organic systems is still poor, and considerably worse than UNIFAC-Dortmund. This shows that, while some information about organic systems is retained from the pre-trained model, the fine-tuning phase mostly guide the model away from the original organic space towards the IL space.

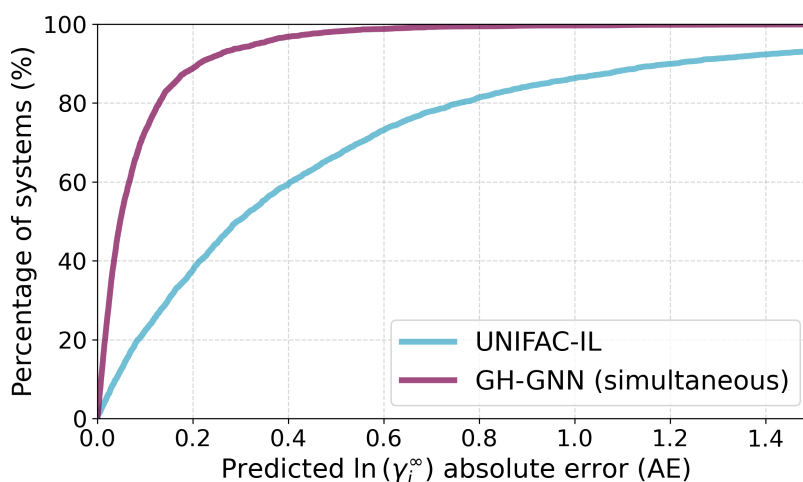


Fig. 4.10.: Cumulative distribution of the absolute prediction error of the extended GH-GNN and the UNIFAC-IL [32] models. The results are shown for the test set.

The third strategy is particularly interesting. The fact that both type of systems are used during the model training, allows the GH-GNN model to simultaneously predict IDACs of organic and IL systems both with remarkable accuracy. In fact, it can be observed that the most general GH-GNN model (i.e., from the third strategy) achieves a higher percentage of systems predicted with an absolute error below 0.3 compared to the GH-GNN model tailored for predicting only individual system types (i.e., the GH-GNN model from Section 4.1). This occurs even when the most general GH-GNN model did not include the multi-task pre-training step as opposed to the original GH-GNN model. From this, one can conclude that GNN-based models highly benefit from having access to an increased amount of data that is also chemically diverse. The reason for this is that the model is presented with more instances of similar molecular interactions besides expanding its realm to new electrostatic ones, which appears to facilitate the learning of data patterns much better.

The GNN and MCM models reported by [132] perform similarly to the general GH-GNN model. However, these methods are constrained to IL systems as opposed to the most general framework developed here. In both organic and IL systems the UNIFAC-based models perform the worst, excluding the specifically tailored GH-GNN instances for organic and IL systems. For illustration, Fig. 4.10 shows the cumulative percentage of systems predicted at various absolute errors for the UNIFAC-IL and the extended GH-GNN models. It is clear that the GH-GNN model is able to predict most systems more accurately than UNIFAC-IL. This is surprising, specially when considering that UNIFAC-IL [32] was, in great part, parameterized from IDAC data. Again, this showcase how the fragmentation and tedious parametrization of UNIFAC-based models can be effectively replaced by the end-to-end approach of GNN-based models (made possible by the advances in automatic differentiation and enhanced computing power that it is available nowadays). This shows the potential that machine learning-based models in general, and GNN-based models in particular, have for developing alternative strategies to the classical predictive thermodynamic methods, which hopefully could contribute to the acceleration towards a sustainable world.

4.3 Extending the GH-GNN model to polymer solutions

In previous Sections, the performance of the proposed hybrid model, GH-GNN, for predicting temperature-dependent IDACs of organic and IL systems was presented. The results suggest that the proposed methodology is able to learn distinct molecular interactions for these types of systems, in general outperforming UNIFAC-based models in the prediction of IDACs. In this Section, the GH-GNN model is extended to yet another realm of relevant mixtures: polymer solutions. Polymer solutions are integral to modern life, notably in plastics production, which has surpassed nearly all other materials in production volume [50]. Predicting IDACs of polymer-small molecule systems is crucial for the advancement of industrial processes towards sustainability, where predictive models can aid in selecting more environmentally friendly compounds [89, 37].

In the realm of polymer solutions, defining molar-based activity coefficients γ becomes ambiguous due to the distribution of a polymer's molecular mass rather than a precise value. Furthermore, the disparity in molecular masses between a polymer j and a smaller molecule i hinders the use of the molar fraction as a concentration unit [110]. Consequently, in light of these challenges, [117] suggested employing a weight fraction activity coefficient, which is defined as

$$\Omega_i = \gamma_i \frac{M_j}{M_i} \quad (4.23)$$

where M refers to the molar mass of the compound. In the present extension of GH-GNN to polymer solutions $\ln \Omega_i^\infty$ was used. To exemplify the usefulness of the weight-based definition of the activity coefficient, one can observe the following expression used for determining IDACs via inverse gas chromatography [117]:

$$\ln \gamma_i^\infty = \ln \left(\frac{273.15 \cdot R}{V_g^0 \cdot M_j \cdot P_i^{sat}} \right) - \frac{P_i^{sat} \cdot (B_i - V_i)}{R \cdot T} \quad (4.24)$$

where, R represents the universal gas constant, T represents the system's absolute temperature, V_g^0 signifies the specific retention volume of the carrier gas corrected to 273.15 K, P_i^{sat} stands for the vapor pressure of the small compound i , and B_i and V_i denote its second-virial coefficient and molar volume, respectively. By introducing Eq. 4.23 into Eq. 4.24, the explicit dependency on the polymer's molar mass M_j is effectively by-passed, enabling the direct measurement of the weight-based IDAC from the retention volume:

$$\ln \Omega_i^\infty = \ln \left(\frac{273.15 \cdot R}{V_g^0 \cdot M_i \cdot P_i^{sat}} \right) - \frac{P_i^{sat} \cdot (B_i - V_i)}{R \cdot T} \quad (4.25)$$

The molecular graphs used in this extension of the GH-GNN model are defined by the same node and edge features given in Tables 4.1 and 4.2, respectively. However, a distinction was made between the polymer graph and the small molecule graph regarding their global-level features vector. While for the small molecule graph the same global-level features, as described in Table 4.3, were used, the vector of global-features for the polymer graph was extended to also accommodate the polymer number average molecular mass M_n and/or the weight average molecular mass M_w depending on the available polydispersity information. Further details on the polymer graph are given in Section 4.3.2. Once again, the same model architecture, hyperparameters, and training specifications as with the original GH-GNN model were maintained for the present extension to polymer solutions. However, to accommodate the distinct dimensions between the initial global-level features of the small molecule graph, $\mathbf{u} \in \mathbb{R}^3$, and that of the polymer graph, $\mathbf{u} \in \mathbb{R}^{\{4,5\}}$, two different initial message-passing layers were used within the molecular-level GNN, one in charge of processing the small molecule graph, and the other the polymer graph.

Tab. 4.9.: Information of the three distinct data subsets used for the extension of the GH-GNN model to polymer solutions.

Information	Data subset		
	M_n	M_w	M_n/M_w
# points	2532	2763	1666
Polymers	42	28	22
Small molecules	137	122	107
% of matrix observed	10.71	16.04	16.19

The extended GH-GNN model presented here was compared to the UNIFAC-based models UNIFAC-ZM [171] and Entropic-FV [90]. These phenomenological models incorporate a free-volume factor or an adjustment in the polymer volume parameter to accommodate the significant variations in molecular sizes inherent in polymer solutions. The performance of the extended GH-GNN model was tested while interpolating among the chemical compounds considered during training, and for the task of extrapolating to other small molecules.

4.3.1 Data set

The data set employed for the extension of the GH-GNN model to polymer solutions comprises experimentally collected $\ln \Omega_i^\infty$ values obtained via inverse gas chromatography, available from Vol. XIV of the DECHEMA Chemistry Data Series [66]. Specifically, only systems involving homopolymers are considered in this study. Through the digitization process of the data using OCR and meticulous manual verification, errors in the original data collection have been detected and rectified, as detailed in Appendix A.14. Data points lacking polydispersity information were eliminated, and repeated measurements were averaged to derive a single data point for the same binary system at the same state conditions.

The resulted curated data set encompasses 48 different homopolymers and 150 small molecules. Some of the data points report either the M_n , the M_w or both values regarding polydispersity. Therefore, three different subsets are constructed, one collecting all data points with available M_n information, the second for the ones having M_w information and the third collecting data points with both polydispersity values. Table 4.9 presents the number of data points, distinct polymers, and distinct small molecules in each data subset, along with the percentage of polymer-small molecule observations relative to all possible combinations of the corresponding polymer-small molecule matrix. It is important to note that, across all three data sets, over 77% of observations correspond to the top 8 most prevalent polymers

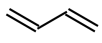
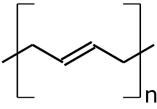
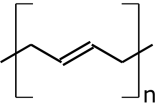
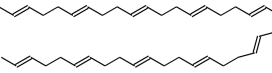



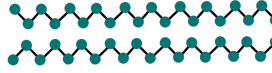
	Monomer	Repeating unit	Periodic unit	Oligomer
Molecular structure				
Graph representation				

Fig. 4.11.: Graph polymer representations used in the extension of the GH-GNN model to polymer solutions.

within each data subset. Similarly, approximately 50% of observations correspond to the top 15 most common small molecules.

4.3.2 Polymer representations

Polymers have been historically represented by their monomer structures on UNIFAC-based models [110, 90, 171]. However, this representation lacks the information about the polymerization points of the molecule, which makes it possible for a single monomer structure to represent multiple polymers [152]. In this study, four primary graph representations have been used and compared: monomer, repeating unit, periodic unit and oligomer. The monomer corresponds to the chemical species used in the synthesis of the polymer (e.g., ethylene glycol for polyethylene glycol). The repeating unit [7] corresponds to the repeated molecular segment in the polymer chain. The periodic unit refers to a special graph representation in which the repeating unit is enlarged with an extra edge between the polymerization points. Finally, the oligomer representation refers to a block of n repeated units, in which n denotes the polymerization degree. In this study, $n = 10$ as it was observed to be a good compromise between computational efficiency (due to the size of the graph) and information difference compared to larger oligomers. Fig. 4.11 schematically illustrates these four polymer representations exemplified by polybutadiene.

4.3.3 Interpolating among systems

To assess the comparative utility of the four graph representations of polymer species, 90% of the polymer-small molecule pairs from each subset were designated for training the GH-GNN model. The remaining data points were allocated for testing,

ensuring the test set solely comprises interpolation cases. Any mixture featuring an unseen polymer or small molecule was reassigned to the training set. This process is iterated 10 times using distinct random seeds to achieve ten distinct train/test splits that are used for a more robust evaluation of the model’s performance. Across all three data sets (i.e., M_n , M_w , and M_n/M_w), the proportion of training points was maintained at an average of 90.5%.

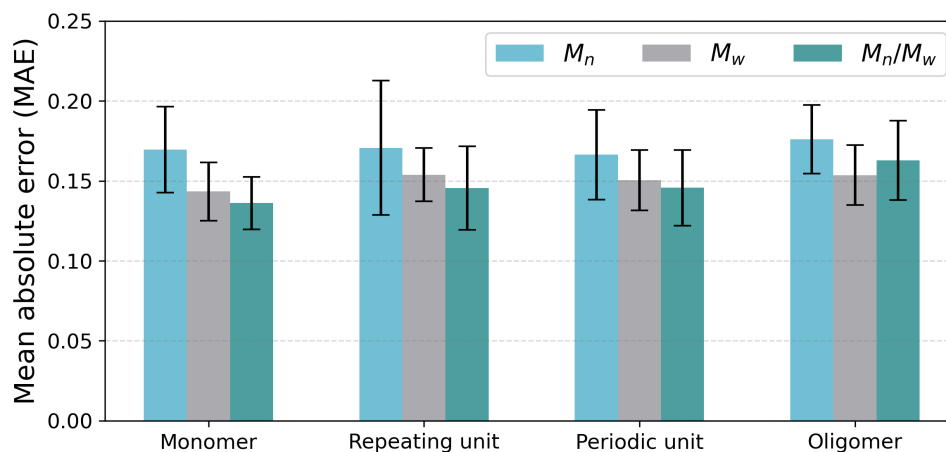


Fig. 4.12.: Mean absolute error (MAE) achieved by the extended GH-GNN trained with different polymer representations on each subset.

Fig. 4.12 shows the MAE achieved by the GH-GNN model on each subset and for each graph representation. The values correspond to the average value of the 10 independent runs and the error bars are determined by the standard deviation of them. As can be noted, no statistical significant difference can be observed when comparing different polymer representations on the same subsets (i.e., comparing bars of same color). The same is true when comparing the different models trained using different polydispersity information (i.e., comparing bars of the same polymer representation). While on average, the models trained using both M_n and M_w perform the best, the performance’s variation of each case stays within each others.

The fact that no significant advantage could be observed among different polymer representations and data subsets is attributed to the specific data set analyzed in this extension. This data set is relatively limited in terms of polymer diversity (cf., Table 4.9), which makes it difficult to distinguish the benefits and pitfalls of each specific representation. This contrasts with the findings reported by [7] on a different (and much more chemically diverse) polymer-property prediction task, which reported that the periodic unit tends to preserve the information much better allowing models to learn the data patterns better. This suggests that in scenarios with limited data availability, the exact polymer representation may not impact the model accuracy

Tab. 4.10.: Interpolation performance achieved by the extended GH-GNN model (with and without pre-training on organic systems) and the random forest baseline using the periodic unit representation. The standard deviation is shown in between parenthesis.

MAE ↓	Interpolation		
	M_n	M_w	M_n/M_w
Random forest	0.27 (0.07)	0.22 (0.03)	0.27 (0.07)
GH-GNN	0.17 (0.03)	0.15 (0.02)	0.15 (0.02)
GH-GNN (pre-trained organic)	0.13 (0.02)	0.13 (0.02)	0.13 (0.03)
R ² ↑	Interpolation		
	M_n	M_w	M_n/M_w
Random forest	0.72 (0.13)	0.79 (0.05)	0.69 (0.17)
GH-GNN	0.90 (0.04)	0.91 (0.03)	0.92 (0.03)
GH-GNN (pre-trained organic)	0.94 (0.02)	0.94 (0.01)	0.94 (0.03)

as much compared to situations with abundant data. However, this observation is contingent upon the inherent limitations of certain polymer representations, as previously discussed.

Table 4.10 shows the performance of the extended GH-GNN model in the interpolation task in terms of the MAE and the R² coefficient. The results are shown for the models trained using the periodic unit. Moreover, the performance of a second GH-GNN model is shown, which uses the original GH-GNN model as a pre-trained step that is later fine-tune using the polymer solution data. For comparison, the performance of a baseline random forest model (100 estimators and expanded to full purity) is shown. This random forest was trained on the concatenated Morgan fingerprints (radius 4 and 2048 bits) of the small molecule and the polymer periodic unit. The temperature and polydispersity information was also included via concatenation.

The GH-GNN model pre-trained on organic systems outperforms the baseline in all metrics and all subsets, and it also outperformed the GH-GNN model trained directly on the polymer solution data. This, showcases the benefits of transfer learning when dealing with limited data. As opposed to the ionic liquid case (i.e., Section 4.2), a simultaneous approach was not attempted here, given the significant disparity in the amount of data available for organic systems compared to that of polymer solutions. It should be also noted that, in almost all cases, the variation of the predictions decrease for the pre-trained model compared to the direct approach. This also shows that transfer learning does not only increases the model's accuracy, but also its robustness.

Tab. 4.11.: Extrapolation performance achieved by the extended GH-GNN model (with and without pre-training on organic systems) and the random forest baseline using the periodic unit representation. The standard deviation is shown in between parenthesis.

MAE ↓	Extrapolation		
	M_n	M_w	M_n/M_w
Random forest	0.26 (0.07)	0.31 (0.05)	0.24 (0.08)
GH-GNN	0.24 (0.14)	0.26 (0.06)	0.18 (0.06)
GH-GNN (pre-trained organic)	0.15 (0.05)	0.20 (0.07)	0.15 (0.05)
R^2 ↑	Extrapolation		
	M_n	M_w	M_n/M_w
Random forest	0.63 (0.23)	0.45 (0.35)	0.66 (0.15)
GH-GNN	0.66 (0.46)	0.64 (0.18)	0.86 (0.08)
GH-GNN (pre-trained organic)	0.90 (0.09)	0.81 (0.14)	0.89 (0.07)

4.3.4 Extrapolating to other small molecules

To assess the GH-GNN model’s performance when extrapolating to unseen small molecules, a list of all small molecules in each subset is initially compiled. Subsequently, a random selection comprising 90% of these unique molecules is conducted. The training set is then constructed, comprising all polymer-small molecule pairs containing any of these selected small molecules, while the testing set consists of the remaining unique mixtures not present in the training set. Similar to the interpolation scenario, the models undergo evaluation via 10 independent splits utilizing different random seeds, with reported metrics representing averages across these 10 splits. On average, the proportion of training points across all three data subsets (M_n , M_w , and M_n/M_w) amounted to 88.9%.

Table 4.11 shows the outcomes obtained for the extrapolation task. An important observation arises when comparing the GH-GNN model’s performance between the interpolation (Table 4.10) and extrapolation (Table 4.11) tasks. It becomes evident that the accuracy of the model diminishes during extrapolation, and the predictions’ variability increases. This trend simply remarks that extrapolating to unknown chemical species is a more challenging task compared to interpolating within the chemical space observed during training. As discussed earlier in this Chapter IDAC interpolation tasks can be even achieved with remarkable accuracy using matrix or tensor completion techniques, which do not necessitate explicit information about the chemical structures and interactions [33, 74, 34]. Similarly to the interpolation task, transfer learning consistently improves the model’s accuracy and robustness.

In order to benchmark the performance of the extended GH-GNN model against the phenomenological models UNIFAC-ZM [171] and Entropic-FV [90], a collection of systems was gathered from the literature [115] classified into athermal systems,

Tab. 4.12.: Comparison between the phenomenological models UNIFAC-ZM [171] and Entropic-FV [90] against the GH-GNN model extended for polymer solutions.

Systems	# points	UNIFAC-ZM	Entropic-FV	GH-GNN (pre-trained organic)
Athermal	53	11.1	9.3	4.0
Polar	66	22.6	11.2	6.4
Associated	21	27.9	33.8	22.3

polar systems and systems with association. The performance of the GH-GNN model corresponds to the ensemble of 10 models trained for extrapolation (for the case including pre-training in organic systems). The performance, as shown in Table 4.12, is measured by the MAPE on the unscaled Ω_i^∞ values. Only systems contained in the test set of each model in the ensemble are included in the comparison. The best performance is shown in bold.

For both phenomenological models, temperature-dependent interaction parameters were used as they were found to perform better than temperature independent ones [115]. For the athermal scenario, the studied systems included polyethylene, low-density polyethylene, and polyisobutylene, alongside linear, branched, and cyclic alkanes. In polar systems, the comparison encompassed poly(ethyl methacrylate), poly(methyl acrylate), poly(methyl methacrylate), poly(n-butyl methacrylate), poly(vinyl acetate), polybutadiene, and polystyrene, with solvents such as ketones, esters, chlorinated hydrocarbons, benzene, and toluene. Concerning systems involving associations, the data includes poly(ethylene oxide), poly(methyl methacrylate), polybutadiene, and polystyrene, along with monohydroxy alcohols and acetic acid. The extended GH-GNN model consistently demonstrated lower MAPE values across all three system types compared to the UNIFAC-based models. This, once again, showcases that hybrid GNN-based models are promising alternatives to construct predictive thermodynamic models across different types of systems achieving state-of-the-art accuracy.

4.4 Chapter summary

This Chapter introduces the Gibbs-Helmholtz Graph Neural Network (GH-GNN) employing a hybridization strategy that combines mechanistic and machine-learning submodels. Initially, a GNN operating at the molecular level learns embeddings directly from molecular graphs, incorporating information on polarity and polarizability via global-level features. Building on this, a mixture graph is constructed to capture hydrogen-bonding information as originally proposed by [125], enhancing the model's capacity to learn mixture representations via a secondary mixture-level

GNN. Utilizing the Gibbs-Helmholtz equation and the assumption of temperature-independent partial molar excess enthalpy, an expression is derived to introduce temperature-dependency into the IDACs.

The GH-GNN undergoes a pre-training phase in a multi-task fashion to predict the temperature-independent parameters of the derived Gibbs-Helmholtz expression before fine-tuning on all available IDAC data. This approach outperforms incorporating temperature as an additional input parameter. Extensive testing demonstrates the model's accuracy across various temperatures and chemical species, particularly within the training mixture space, surpassing the UNIFAC-Dortmund model [59]. However, performance outside the training mixture space is influenced by chemical proximity, quantified through the proposed Jaccard distance metric. Notably, the GH-GNN exhibits superior prediction accuracy compared to state-of-the-art UNIFAC models in organic systems (UNIFAC-Dortmund [59]), ionic liquids (UNIFAC-IL [32]), and polymer solutions (UNIFAC-ZM [171] and Entropic-FV [90]), highlighting the potential of GNN-based models for predictive thermodynamic modeling, particularly in temperature-dependent IDAC predictions.

In addition to the significant enhancements in IDAC prediction accuracy achieved by the proposed GH-GNN model over UNIFAC-based models, it is crucial to underscore an additional advantage. The implementation time for constructing GNN-based models tailored to specific prediction tasks is (arguably) considerably shorter compared to the development of traditional group-contribution strategies. The main reasons for this is that the recent advances in automatic differentiation, open-sourced deep learning frameworks and modern GPU computing allow for a fast implementation of end-to-end solutions by-passing the previously manual task of molecular fragmentation followed by the model parametrization. In this same spirit, additional efforts were dedicated for the collection, digitization and cleaning of data sources that allowed for the open-sourcing of the models presented and investigate in this Chapter.

Predicting Activity Coefficients

” *...one doesn't bet against deep learning.*

— Ilya Sutskever
Computer scientist

In Chapter 3 the simplest case of predicting isothermal IDACs using hybrid GNN-based models was explored. Following, Chapter 4 extended the case for predicting temperature-dependent IDACs. In this Chapter, the next logical step is taken: introducing the composition dependency to predict the most general activity coefficients γ_i .

Similarly to the Gibbs-Helmholtz equation (Eq. 2.25) that establishes the temperature dependency of activity coefficients, there exist a thermodynamic relationship that constraints the composition dependency. This relationship is the Gibbs-Duhem equation (Eq. 2.27), which establishes that the activity coefficients of the components in a given mixture are not independent from each other. For binary systems, the Gibbs-Duhem relation can be written as

$$0 = x_i d \ln \gamma_i + x_j d \ln \gamma_j \quad (5.1)$$

where, x refers to the molar fraction of the corresponding species.

This Chapter explores the extension of the proposed Gibbs-Helmholtz Graph Neural Network (GH-GNN) to include the composition dependency of activity coefficients ensuring thermodynamic consistency. Moreover, it explores the performance of the proposed general framework when predicting activity coefficients of binary systems in vapor-liquid equilibrium (VLE) conditions. A comprehensive comparison to the UNIFAC-Dortmund model is also provided. Additionally, some insights into the performance of the proposed model for predicting VLEs of ternary mixtures are included.

5.1 From infinite to finite using the Margules equation

One has to bear in mind that activity coefficients can be calculated from the molar excess Gibbs energy of the mixture (cf. Eq. 2.29). This implies that if an expression for this excess property is available, all activity coefficients of the mixture components can be computed. However, the concept of excess Gibbs energy, as briefly discussed in Section 2.1.1, is founded on the notion of refining/correcting the predictions of an ideal model, specifically the ideal solution model (Raoult's law). Hence, rather than being an intrinsic physical property of matter, the molar excess Gibbs energy (and consequently the activity coefficients) represents a theoretical construct aimed at modeling the physical behavior of condensed matter. Consequently, various expressions can be proposed to model this function, known as g^E models. The only requirement for such expressions is that they must satisfy the limiting condition outlined in Eq. 2.30.

The simplest g^E model for the simplest mixture (i.e., binary mixture) that can be proposed corresponds to the Porter model [56]

$$g^E = Ax_i x_j \quad (5.2)$$

where, the parameter A should be fitted to experimental data.

However, a more general framework can be constructed by considering the extended Margules equation [108], which approximates the g^E function using a p th-order Taylor polynomial. This model can be applied to mixtures of n components (complete derivation in Appendix A.15). For binary systems the g^E Margules model can be expressed as follows:

$$g^E = x_i x_j (x_i w_{ji} + x_j w_{ij}) \quad (5.3)$$

As shown in Appendix A.15, all the coefficients of the Margules model (e.g., w_{ij} and w_{ji} in Eq. 5.3) can be fully determined by the binary-system IDACs of all pair combinations of the mixture components. The g^E Margules model not only satisfies the boundary condition of excess properties (Eq. 2.30), but also is consistent with the Gibbs-Duhem relationship (Eq. 2.27). For a binary-mixture, the extended Margules equation computes the activity coefficients as

$$\ln \gamma_i = 2w_{ji}x_i x_j + x_j^2 w_{ij} - 2g^E \quad (5.4)$$

$$\ln \gamma_j = 2w_{ij}x_j x_i + x_i^2 w_{ji} - 2g^E \quad (5.5)$$

$$w_{ji} = \ln \gamma_j^\infty \quad (5.6)$$

$$w_{ij} = \ln \gamma_i^\infty \quad (5.7)$$

Therefore, by incorporating the extended Margules equation into the GH-GNN framework developed in Chapter 4, thermodynamic consistent activity coefficients at varying temperatures and compositions can be predicted as soon as the corresponding IDACs are available (in this case predicted via the GH-GNN model from Chapter 4). In this way, the hybrid serial model GH-GNN is here further extended by including the phenomenological Margules model also in a sequential manner. Fig. 5.1 shows a schematic representation of the proposed hybrid serial model for predicting activity coefficients. In the following Sections, the performance of such GH-GNN-Margules hybridization strategy is discussed when predicting fluid phase equilibria.

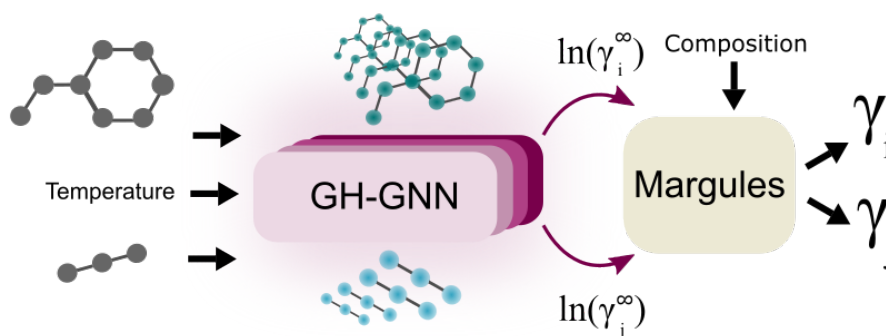


Fig. 5.1.: Schematic illustration of the proposed hybrid serial model consisting of the GH-GNN model and the extended Margules equation for predicting activity coefficients.

5.2 Predicting binary vapor-liquid equilibria

The accurate prediction of VLE behavior is important for designing and modeling thermal separation processes, particularly evident in applications such as distillation columns, which are the most popular separation technique in the chemical industry [103]. Given its widespread usage, distillation has undergone extensive study, with VLE data being the most measured equilibrium property of mixtures [56]. However,

despite decades of research, the available VLE data remains limited, with only a small fraction, estimated at merely 1.2% for binary systems of interest being experimentally measured [56]. Data for mixtures exceeding two components is even scarcer. Hence, the significance of predictive methods, such as the one proposed in this work, that leverage binary data exclusively, becomes apparent. The GH-GNN-Margules approach introduced in this Chapter further refines its requirements by relying solely on experimental binary-system data at infinite dilution for select mixtures.

The VLE of systems with a vapor phase that behaves close to ideality can be model using Eq. 2.23. In this expression the activity coefficient can be approximated directly from the experimental VLE data by

$$\gamma_i = \frac{y_i P}{x_i P_i^{sat}} \quad (5.8)$$

Therefore, by estimating the activity coefficients using the proposed GH-GNN-Margules approach, and by having access to the vapor pressure of the mixture components, one can solve Eq. 5.8 to estimate the VLE behavior of a mixture directly from the binary IDAC information. This approach is explored in this Chapter for the cases of binary isothermal and isobaric VLEs. Additionally, some insights on the performance of the proposed model for predicting VLEs of mixtures containing ionic liquids and ternary VLEs are provided. Since, in all these cases the vapor phase is assumed to be ideal, the discussion and conclusions are limited to systems under low to medium pressures (approx. ≤ 500 kPa).

5.2.1 Data set

For testing the performance of the proposed GH-GNN-Margules hybrid model a collection binary VLE experimental data was gathered from the Korean Data Bank [91]. A cleaning pre-processing step was carried out for filtering out data points containing typographic errors (e.g., misplacement of the decimal point, composition values greater than 1). The standardization of the measurement units of all experimental subsets was also carried out. Only systems at a pressure of 500 kPa or less were considered, as this supports the assumption of ideal vapor phase.

Moreover, pure component data (in specific CAS-RN number and vapor pressure correlation coefficients) were gathered from the Korean Data Bank [91]. SMILES strings were collected from *PubChem* [123] using the CAS-RN identifiers and conserving

Tab. 5.1.: General information of the KDB data set consisting of binary VLE experimental measurements.

Description	Information
Number of data points	26,323
Number of compounds	213
Number of unique binary combinations	887
Number of isobaric subsets	745
Number of isothermal subsets	852
Number of random subsets	28
Pressure range (kPa)	0.01 - 499.50
Temperature range (K)	255 - 576.93

the isomeric information in case isomeric SMILES were available. For compounds without CAS-RN identifier, the SMILES were retrieved using the name of the compound from *OPSiN* [111]. All systems with compounds that do not have available vapor pressure information were discarded. Also, systems at a temperature outside the applicability bounds of the available vapor pressure correlation parameters were discarded. The vapor pressure was estimated using the Korean Data Bank correlation

$$\ln(P_i^{sat}) = A_i \ln(T) + \frac{B_i}{T} + C_i + D_i T^2 \quad (5.9)$$

where, the pressure is given in kPa and the temperature in K. Furthermore, only data points with reported liquid and the corresponding vapor molar fractions were included. Systems with compounds with atomic or bond features different than the ones considered by the GH-GNN model (cf. Tables 4.1 and 4.2, respectively) were discarded.

The compounds contained in the KDB data set are classified into specific chemical classes available also at the Korean Data Bank [91]. This classification was used here to study the performance of the GH-GNN-Margules framework across different mixtures of chemical classes. To constrain the application of the methodology to systems where the GH-GNN is expected to produce relative accurate estimations of the IDACs, a Jaccard distance metric threshold of 0.6 (as mentioned in Section 4.1.5) was used to filter-out systems outside of the applicability domain of the model. In total, 1,963 data points were discarded due to having a high Jaccard distance metric. A summary of the information of the cleaned data set, here referred to as the *KDB data set*, is presented in Table 5.1.

5.2.2 Isothermal vapor-liquid equilibria

In the case of isothermal systems contained in the KDB data set, the number of data points reduces to 12,427, covering measurements of 166 compounds in 503 unique binary combinations. The temperature range covered goes from 260.93 to 573.15 K. As shown in Equations. 5.4 to 5.7, for computing the activity coefficients of a binary mixtures using the extended Margules equation, the two corresponding IDACs are necessary. Therefore, the number of IDACs necessary for calculating the isothermal VLEs of the 503 binary combinations is 1,006. For 120 (23.8%) binary systems, out of the total 503, both IDACs were observed during training. For 360 (71.6%) systems, the GH-GNN model has to perform an interpolation for at least one of the two IDACs needed, and for the rest, 23 (4.6%) systems, the GH-GNN would need to extrapolate for at least one of the two necessary IDACs. The latter is a result of having 12 compounds (out of the 166 compounds contained in the isothermal VLE data) that were not present at all during the training of GH-GNN. This discussion shows that, even when the extended Margules equation can be (and should be, when possible) applied directly to experimental IDACs, the small availability of experimental measurements is still a significant challenge and a strong motivation for developing predictive methods.

Algorithm 1: Isothermal vapor-liquid equilibrium of binary mixture using the GH-GNN-Margules model

Data: System's temperature, vapor pressure parameters for components i and j , parameters $K_{1,ij}$, $K_{2,ij}$ and $K_{1,ji}$, $K_{2,ji}$ from the GH-GNN model, molar fraction x_i in the liquid phase.

Result: Molar fraction y_i in the vapor phase.

- 1 Compute $w_{ij} = \ln \gamma_{ij}^\infty$ and $w_{ji} = \ln \gamma_{ji}^\infty$ using Eq. 4.2;
 - 2 Compute γ_i and γ_j using Eqs. 5.4 - 5.5;
 - 3 Compute vapor pressures P_i^{sat} and P_j^{sat} using Eq. 5.9 at the system's temperature ;
 - 4 Compute partial pressures using Eq. 5.8 (e.g., $p_i = x_i \gamma_i P_i^{sat}$) ;
 - 5 Compute system's pressure as $P = p_i + p_j$;
 - 6 Compute molar fraction of i in the vapor phase as $y_i = P/p_i$
-

Algorithm 1 shows the procedure to compute isothermal vapor-liquid equilibria of binary mixtures using the proposed GH-GNN-Margules model. Fig. 5.2 shows the performance of the proposed GH-GNN-Margules framework when predicting the isothermal KDB data (according to the predicted molar fraction in the vapor phase) of all different binary chemical classes. It can be seen that overall, the GH-GNN-Margules model, which only uses information at infinite dilution, is able to predict various types of systems with significant accuracy. For illustration, Fig. 5.3 shows the

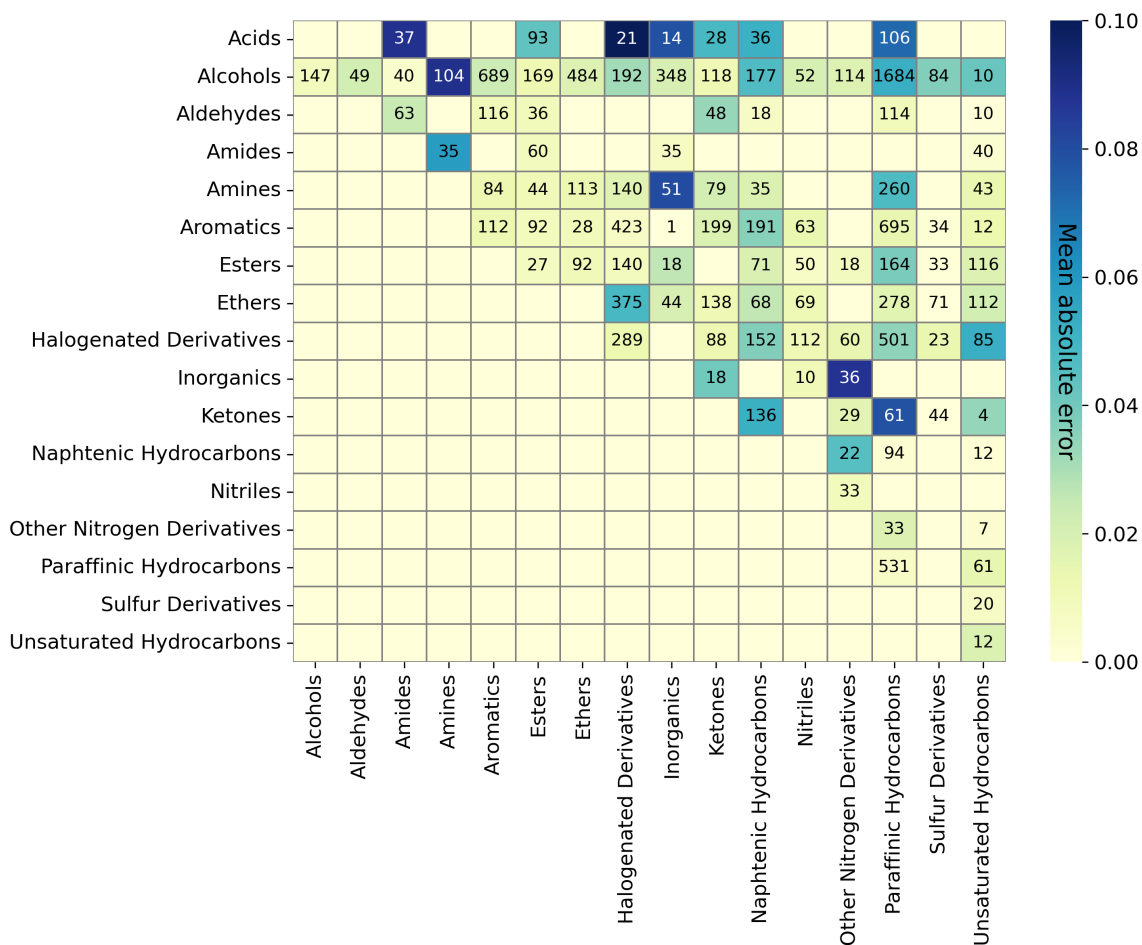


Fig. 5.2.: Heatmap of the mean absolute error achieved by the GH-GNN-Margules model on all KDB isothermal data according to binary chemical classes. The number on each cell indicates the number of data points in each subset.

percentage of binary classes that are predicted within different mean absolute error thresholds for systems that can be also predicted by UNIFAC-Dortmund. Around 60% of the 98 binary classes are predicted with a MAE less or equal to 0.02 when using GH-GNN-Margules.

In general, systems containing acids are difficult to predict for the GH-GNN-Margules model, specially in combination with halogenated species and amides. This can be attributed to the fact that these types of species participate in strong hydrogen-bonding and dipole-dipole interactions that could still be difficult to model with the information given to the GH-GNN model or that might require higher order polynomials on the Margules expression. Alcohols/Amines systems also appear to be predicted with poor accuracy. Moreover, systems containing water (categorized within the "Inorganic" class in Fig. 5.2) together with amines and other nitrogen

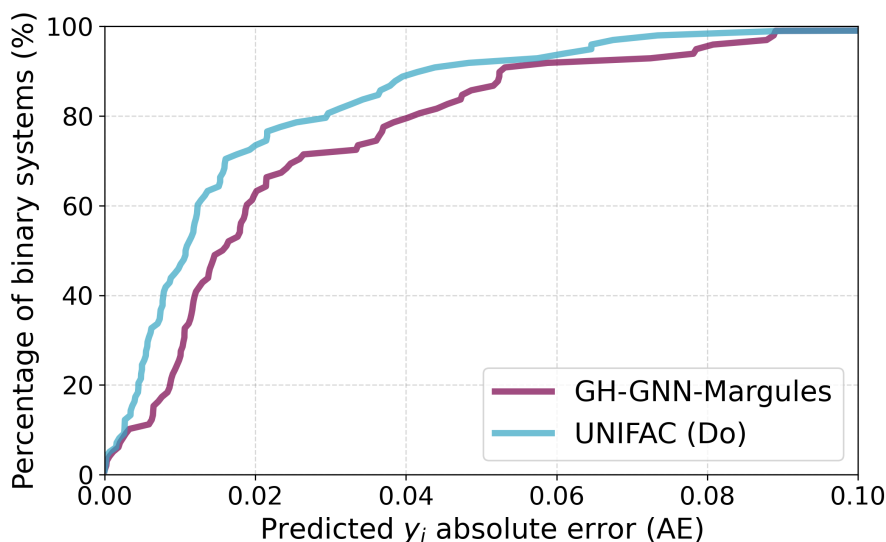


Fig. 5.3.: Cumulative percentage of binary classes predicted by the GH-GNN-Margules and UNIFAC-Dortmund models within different mean absolute error thresholds. The errors are calculated according to the predicted molar fraction in the vapor phase on the isothermal vapor-liquid equilibria KDB data. Only feasible systems for UNIFAC-Dortmund are considered.

derivatives also were predicted with relatively large errors. By contrast, systems containing esters, ethers or aldehydes tend to be well-predicted. Aromatics and unsaturated hydrocarbons are also observed to be predicted with relatively high accuracy.

Overall, the model is able to predict, as shown in Table 5.2, 76.04% of all isothermal data points with an absolute error of less than 0.03. Table 5.2 also categorizes the performance based on the systems that were predicted from entirely observed IDACs (denoted with “Observed”), from interpolated IDACs (denoted with “Interpolated”) and from extrapolated IDACs (denoted with “Extrapolated”). However, one should

Tab. 5.2.: GH-GNN-Margules and UNIFAC-Dortmund performance on predicting isothermal binary vapor-liquid equilibria data. The metrics are shown with respect to the predicted molar fraction in the vapor phase.

	# points	# binary classes	MAE ↓	R ² ↑	AE ≤ 0.03 ↑
All data	12,427	98	0.0277	0.962	76.04%
Observed data	3,588	39	0.0341	0.943	69.43%
Interpolated data	8,265	92	0.0250	0.966	79.23%
Extrapolated data	574	13	0.0258	0.985	71.43%
Feasible systems for UNIFAC (Do)					
UNIFAC (Do)			0.0192	0.968	87.37%
GH-GNN-Margules	12,231	98	0.0278	0.961	75.96%

bear in mind that, in fact, the GH-GNN-Margules model is extrapolating in all cases from the infinite dilution regime to the finite concentration space.

It is interesting to note that the performance of the GH-GNN-Margules model for systems with all necessary IDACs being observed is slightly worse than for systems where interpolation is necessary. This is attributed to the difference between the number of binary classes covered for each case. While “observed” systems involve only 39 binary classes, the “interpolated” systems covered 92. While for some type of binary classes the GH-GNN-Margules model can predict accurate results (cf. Fig. 5.2), for others the performance is relatively poorer.

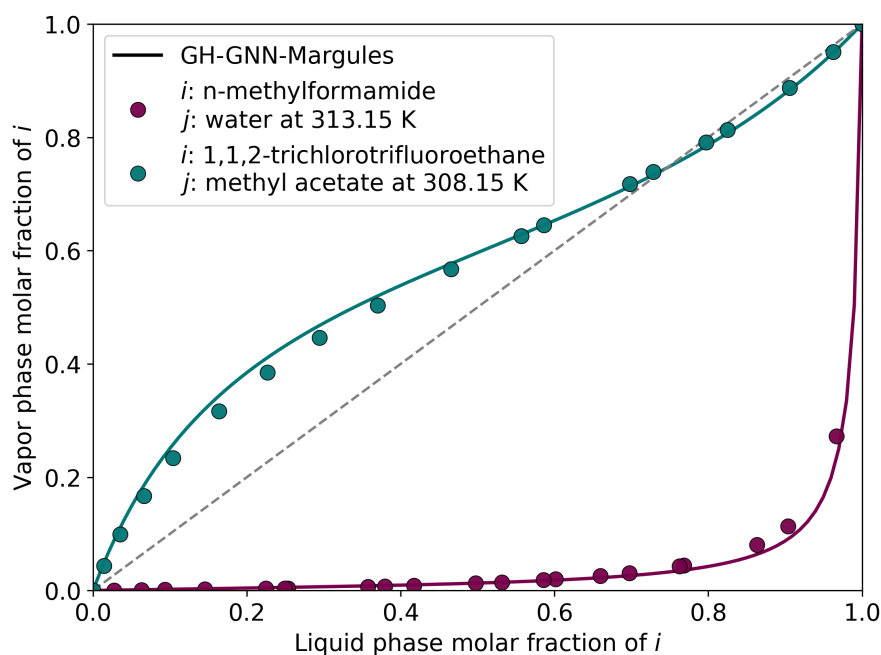


Fig. 5.4.: Isothermal vapor-liquid equilibria diagram of two systems that could not be predicted with UNIFAC-Dortmund, but were predicted by GH-GNN-Margules. Markers denote experimental measurements.

The comparison between the UNIFAC-Dortmund and the GH-GNN-Margules model, shown in Table 5.2, indicates that UNIFAC-Dortmund predicts most of the isothermal VLE systems more accurately than the proposed model. This same conclusion can be drawn by looking at the cumulative percentage of binary systems that are predicted within different MAE errors (as illustrated in Fig. 5.3). Perhaps, this result does not come as a surprise given that extensive VLE data was used for parametrizing UNIFAC-Dortmund [59]. By contrast, the GH-GNN-Margules model achieves such performance by utilizing only experimental data in the infinitely diluted region. And, as a predictive method, the GH-GNN part of the proposed methodology is able to

predict all necessary IDACs for the extended Margules equation to estimate the finite concentration behavior of the mixture.

Moreover, as already discussed in Chapters 3 and 4, many systems simply cannot be predicted by UNIFAC-Dortmund due to missing interaction parameters. In this isothermal VLE study, this type of systems are limited but still present (cf. 12,231 feasible points out of the 12,427 points in Table 5.2). As an example, Fig. 5.4 shows the vapor-liquid equilibria diagram for two of the systems that UNIFAC-Dortmund is not able to predict. Both of these systems correspond to “interpolated” systems for the GH-GNN-Margules model. It is also remarkable that, as shown for the system “1,1,2-trichlorotrifluoroethane / methyl acetate” in Fig. 5.4, the GH-GNN-Margules model is able to capture the azeotropic behavior with excellent precision without including this type of experimental data, as opposed to UNIFAC-Dortmund which was also parameterized using dedicated azeotropic data [59].

5.2.3 Isobaric vapor-liquid equilibria

From the KDB dataset, 13,528 data points are given at isobaric conditions. These data points involve 148 distinct compounds arranged in 501 different binary systems. The pressure in this isobaric subset ranges from 1.33 to 496.63 kPa. The number of systems where both IDACs were observed during the training of the GH-GNN model is 106 (21.2%). Similarly, the number of systems where the GH-GNN model has to perform IDAC interpolation is 388 (77.4%), and for only 7 (1.4%) systems an extrapolation is necessary because 5 (out of the 148) compounds were not present at all during the training of GH-GNN.

When calculating isobaric VLEs an optimization problem has to be solved for minimizing the difference between the calculated system’s pressure and the actual system’s pressure. The optimization problem is here implemented as

$$\begin{aligned}
 \min_T \quad & |P - \hat{P}(T)| \\
 \text{s.t.} \quad & \hat{P}(T) = x_i \gamma_i(T) P_i^{sat}(T) + x_j \gamma_j(T) P_j^{sat}(T) \\
 & \{\gamma_i(T), \gamma_j(T)\} \leftarrow \text{Model}(T) \\
 & T_{\min} \leq T \leq T_{\max}
 \end{aligned} \tag{5.10}$$

where, P and \hat{P} stand for the actual and predicted system’s pressure, Model refers to the specific model being used to compute the activity coefficients (in this case the GH-GNN-Margules or UNIFAC-Dortmund), and T_{\min} and T_{\max} refer to the optimization

bounds for the temperature. In this case, the temperature bounds are determined as the minimum temperature range in which the vapor pressure correlation (Eq. 5.9) remains feasible.

Algorithm 2 shows how to compute isobaric VLE calculations of binary mixtures using the proposed GH-GNN-Margules model. The *while* loop in Algorithm 2 is essentially solving the Optimization Problem 5.10. In this work, this is implemented using the `SciPy` library [158] with the Brent's method using maximum 2,000 iterations and a tolerance of 1.48×10^{-8} .

Algorithm 2: Isobaric vapor-liquid equilibrium of binary mixture using the GH-GNN-Margules model

Data: System's pressure, vapor pressure parameters for components i and j , parameters $K_{1,ij}$, $K_{2,ij}$ and $K_{1,ji}$, $K_{2,ji}$ from the GH-GNN model, molar fraction x_i in the liquid phase, temperature bounds T_{min} and T_{max} .

Result: Molar fraction y_i in the vapor phase.

```

1 Take a random initial  $T$  guess;
  while  $|P - \hat{P}(T)| > \epsilon$  do
2   Compute  $w_{ij} = \ln \gamma_{ij}^\infty$  and  $w_{ji} = \ln \gamma_{ji}^\infty$  using Eq. 4.2;
3   Compute  $\gamma_i$  and  $\gamma_j$  using Eqs. 5.4 - 5.5;
4   Compute vapor pressures  $P_i^{sat}$  and  $P_j^{sat}$  using Eq. 5.9 at the system's
      temperature ;
5   Compute partial pressures using Eq. 5.8 (e.g.,  $p_i = x_i \gamma_i P_i^{sat}$ ) ;
6   Estimate system's pressure as  $\hat{P} = p_i + p_j$  ;
      if termination condition is not met then
7     | Estimate next  $T$  candidate using the Brent's method;
      end
  end
end

```

Table 5.3 shows the performance of the GH-GNN-Margules model in the isobaric data. Similarly to Section 5.2.2, the results are shown for all isobaric systems and for systems where the necessary IDACs were observed, interpolated or extrapolated. Additionally, the comparison to UNIFAC-Dortmund is provided. As with the isothermal systems, UNIFAC-Dortmund outperforms the GH-GNN-Margules model for predicting isobaric VLEs of binary systems. As explained in the previous Section this is expected given the difference in available experimental information that both models had during training.

Fig. 5.5 shows the MAE that the GH-GNN-Margules model achieves on predicting the molar fraction in the vapor phase for each binary combination of chemical classes. The first thing to note is that, compared to the isothermal data, different binary classes are now considered. For instance, data for the combinations acids/alcohols

Tab. 5.3.: GH-GNN-Margules and UNIFAC-Dortmund performance on predicting isobaric binary vapor-liquid equilibria data. The metrics are shown with respect to the predicted molar fraction in the vapor phase.

	# points	# binary classes	MAE ↓	R ² ↑	AE ≤ 0.03 ↑
All data	13,528	72	0.0339	0.950	66.41%
Observed data	3,551	30	0.0294	0.967	67.73%
Interpolated data	9,789	66	0.0353	0.945	66.03%
Extrapolated data	188	7	0.0443	0.924	61.17%
Feasible systems for UNIFAC (Do)					
UNIFAC (Do)	13,512	72	0.0270	0.969	82.55%
GH-GNN-Margules			0.0339	0.950	66.37%

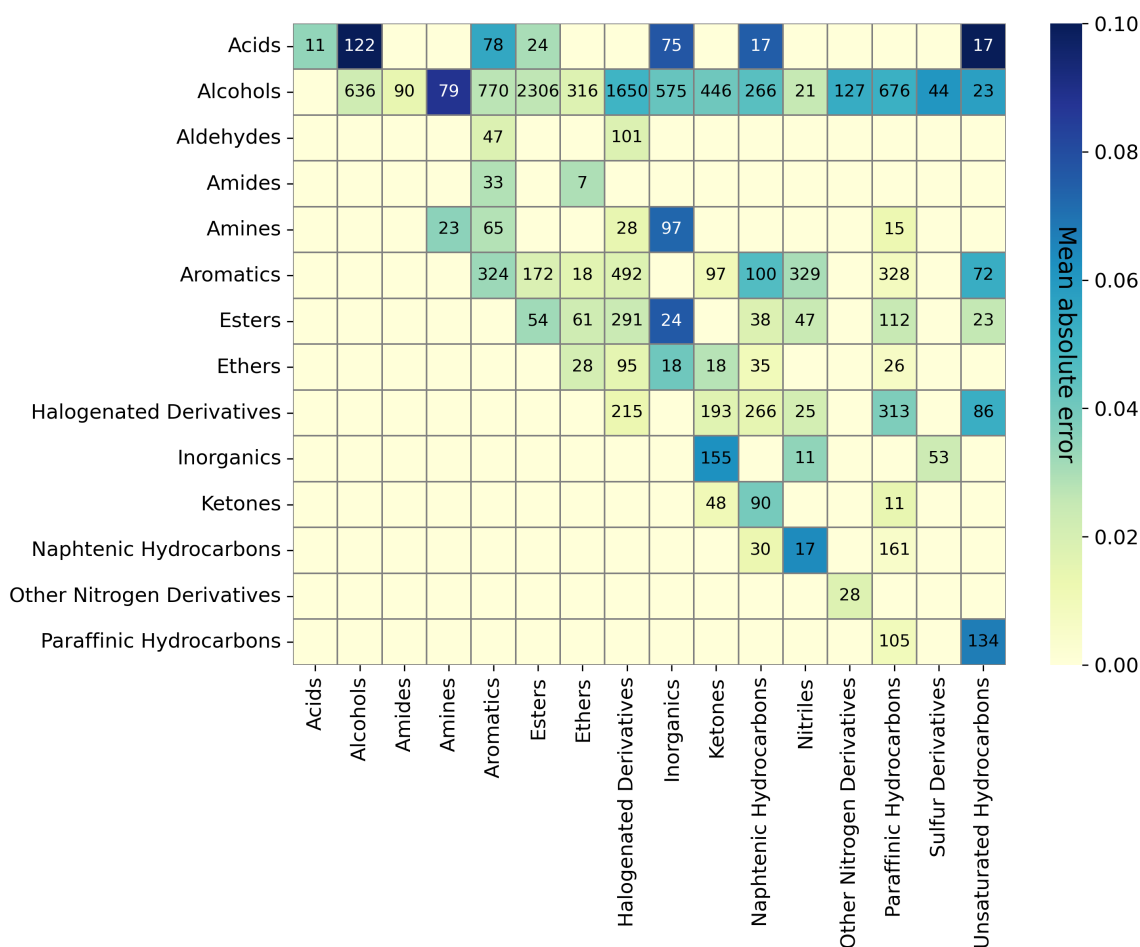


Fig. 5.5.: Heatmap of the mean absolute error achieved by the GH-GNN-Margules model on all KDB isobaric data according to binary chemical classes. The number on each cell indicates the number of data points in each subset.

and acids/acids is available at isobaric conditions, but not at isothermal conditions. In total, 14 new binary classes are present in the isobaric data that were not present in the isothermal data. As with the isothermal case, many binary classes at isobaric

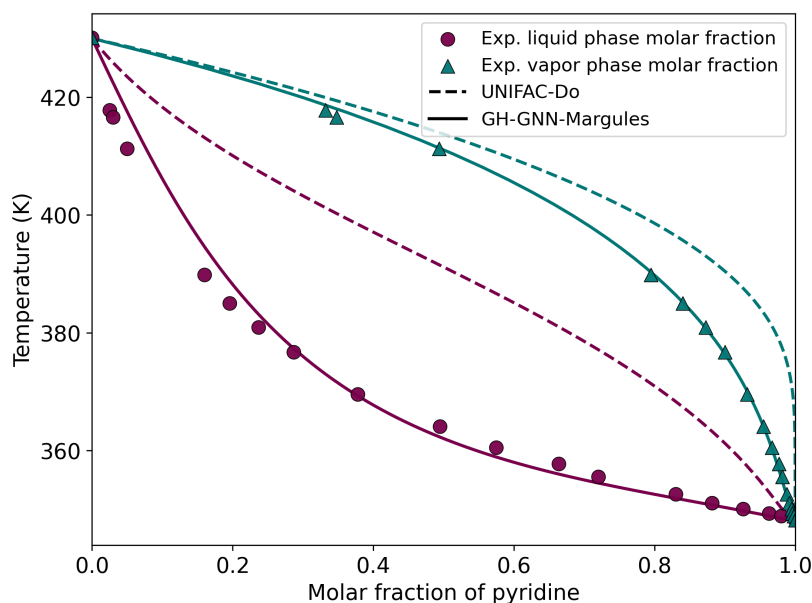


Fig. 5.6.: Isobaric vapor-liquid equilibria diagram of system “pyridine/1,2,3,4-tetrahydronaphthalene” at 26.66 kPa predicted with GH-GNN-Margules and UNIFAC-Dortmund.

conditions are predicted with relatively low errors. In fact, out of the 745 isobaric systems considered here, 175 systems are on average better predicted with the GH-GNN-Margules model rather than with UNIFAC-Dortmund. To illustrate this Fig. 5.6 shows the VLE diagram for the system “pyridine/1,2,3,4-tetrahydronaphthalene” at 26.66 kPa. It is clear that the GH-GNN-Margules model predicts the VLE behavior closer to the experimental observations compared to UNIFAC-Dortmund. The same can be observed in the second example shown in Fig. 5.7, for the system “tetrachloroethylene/furfural” at 101.325 kPa. For both of these systems, the GH-GNN model has to interpolate at least one of the two IDACs needed. This shows that, despite UNIFAC-Dortmund outperforming the proposed model in most cases, counter-examples can also be found.

It is important to highlight that, since the parameters $K_{1,i}$ and $K_{2,i}$ of the GH-GNN model are temperature independent, they need to be predicted only once for each binary combination. As a result, the optimization problem that needs to be solved for isobaric cases (cf. Algorithm 2), is computationally cheaper compared, for instance, to approaches where the temperature dependence is part of the highly parametric deep learning model (e.g., models in [132, 165]).

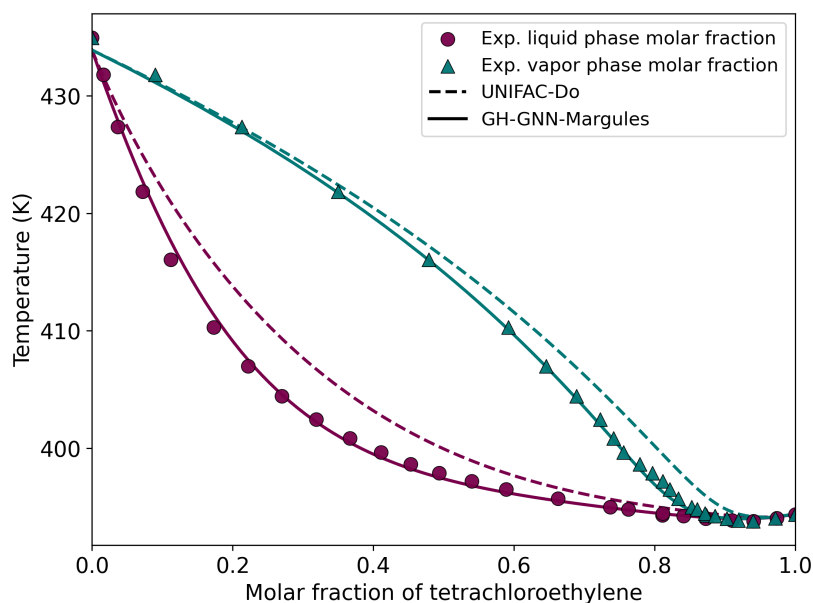


Fig. 5.7.: Isobaric vapor-liquid equilibria diagram of system “tetrachloroethylene/furfural” at 101.325 kPa predicted with GH-GNN-Margules and UNIFAC-Dortmund.

Tab. 5.4.: GH-GNN-Margules and UNIFAC-Dortmund performance on predicting all binary vapor-liquid equilibria data contained in the KDB data set. The metrics are shown with respect to the predicted molar fraction in the vapor phase.

	# points	# binary classes	MAE ↓	R ² ↑	AE ≤ 0.03 ↑
All data	26,323	112	0.0302	0.957	71.76%
Observed data	7,264	44	0.0314	0.954	69.01%
Interpolated data	18,277	108	0.0297	0.957	72.95%
Extrapolated data	782	17	0.0287	0.977	69.56%
Feasible systems for UNIFAC (Do)					
UNIFAC (Do)			0.0206	0.966	85.27%
GH-GNN-Margules	26,089	112	0.0302	0.957	71.71%

5.2.4 Overall performance on binary vapor-liquid equilibria

In order to provide a more comprehensive overview of the performance of GH-GNN-Margules in predicting binary VLEs, the prediction results on the complete KDB data set are provided in Table 5.4. This includes all isothermal and isobaric systems, and all random measurements. The performance on the subset of systems that are feasible for UNIFAC-Dortmund is also provided.

Moreover, to have a more detailed overview of what are the few binary classes that GH-GNN-Margules tends to predict better than UNIFAC-Dortmund, the best model per binary combination is shown in Fig. 5.8. The performance of each binary class is taken as MAE across all data points of the same binary class. The predictions are

Acids	11	122		37		78	102		21	89	28	53			106		17
Alcohols		783	49	105	183	1468	2529	800	1843	955	583	443	73	241	2391	128	33
Aldehydes				63		163	36		101		48	18			114		10
Amides					35	33	60	7		14							40
Amines					23	149	44	113	168	148	79	35			275		43
Aromatics						518	264	46	915	1	296	291	392		1037	34	84
Esters							100	153	412	42		109	97	18	276	33	139
Ethers								28	431	62	156	103	69		304	71	112
Halogenated Derivatives									495		259	418	137	60	792	23	171
Inorganics											173		21	36		53	
Ketones											48	226		29	83	44	4
Naphtenic Hydrocarbons												30	17	22	255		12
Nitriles														33			
Other Nitrogen Derivatives														28	33		7
Paraffinic Hydrocarbons															636		229
Sulfur Derivatives																	20
Unsaturated Hydrocarbons																	12

Fig. 5.8.: Matrix of binary classes contained in the KDB data set that are feasible to predict with UNIFAC-Dortmund. The color shows whether UNIFAC-Dortmund or GH-GNN-Margules achieve a lower mean absolute error (MAE). The number on each cell indicates the number of data points in each subset.

measured with respect to the molar fraction in the vapor phase. Once again, it is evident that the UNIFAC-Dortmund model performs better for most binary classes. However, some binary classes appear to be better predicted with GH-GNN-Margules. For instance, the binary class "amines/aromatics", which includes the system shown in Fig. 5.6, tends to be better predicted with the proposed model. Nevertheless, it is important to highlight that the comparison here is limited to the overall MAE of the predictions. Other important criteria should ideally also be considered based on its relative relevance in applications to process engineering (e.g., the accuracy on predicting azeotropes).

5.3 Predicting ternary vapor-liquid equilibria

The extended Margules equation is not limited to binary-systems, it can also be applied to multi-component systems. For the case of ternary systems it is written as

$$\ln \gamma_i = 2(x_i x_j w_{ji} + x_i x_k w_{ki}) + x_j^2 w_{ij} + x_k^2 w_{ik} + x_j x_k c_{ijk} - 2g^E \quad (5.11)$$

$$\ln \gamma_j = 2(x_j x_k w_{kj} + x_j x_i w_{ij}) + x_k^2 w_{jk} + x_i^2 w_{ji} + x_k x_i c_{jki} - 2g^E \quad (5.12)$$

$$\ln \gamma_k = 2(x_k x_i w_{ik} + x_k x_j w_{jk}) + x_i^2 w_{ki} + x_j^2 w_{kj} + x_i x_j c_{kij} - 2g^E \quad (5.13)$$

$$g^E = x_i x_j (x_j w_{ij} + x_i w_{ji}) + x_i x_k (x_k w_{ik} + x_i w_{ki}) \\ + x_j x_k (x_k w_{jk} + x_j w_{kj}) + x_i x_j x_k c_{ijk} \quad (5.14)$$

$$c_{abc} = \frac{1}{2}(w_{ab} + w_{ba} + w_{ac} + w_{ca} + w_{bc} + w_{cb}) - w_{abc} \quad (5.15)$$

$$w_{ab} = \ln \gamma_{ab}^\infty \quad (5.16)$$

$$(5.17)$$

where, the subscripts a , b and c correspond to the different components in the mixture (either i , j or k) according to the specific case. In this work, the resulting ternary interaction parameter w_{ijk} is assumed to be zero. However, notice that the ternary parameter c_{abc} is non-zero, but determined completely from the binary IDACs.

Tab. 5.5.: Ternary vapor-liquid equilibria data used for evaluating the performance of the GH-GNN-Margules model.

Comp. 1	Comp. 2	Comp. 3	# points	Constant state	Ref.
Acetone	Chloroform	Methanol	71	101.325 kPa	[70]
Hexane	Benzene	Sulpholane	14	101.325 kPa	[126]
Benzene	Heptane	Dimethylformamide	50	101.325 kPa	[21]
Benzene	Heptane	Acetonitrile	12	101.325 kPa	[155]
Acetone	Chloroform	Benzene	53	101.325 kPa	[84]
Benzene	Cyclohexane	Hexane	108	101.325 kPa	[130]
Acetone	Tetrachloromethane	Benzene	57	101.325 kPa	[149]
Ethanol	Benzene	Heptane	50	53.329 kPa	[109]
Hexane	Methanol	Acetone	54	313.15 K	[112]
Chloroform	Methanol	Benzene	70	101.325 kPa	[92]

A collection of 10 distinct ternary VLEs were collected from the literature to get some insights for the performance of GH-GNN-Margules in mixtures of more than two components. Table 5.5 shows some relevant information of the specific systems

studied here. In these mixtures, several chemical classes are represented. For example, aromatics, paraffinic hydrocarbons, nitriles, ketones, alcohols and halogenated and sulfur derivatives. In total, 539 data points were evaluated.

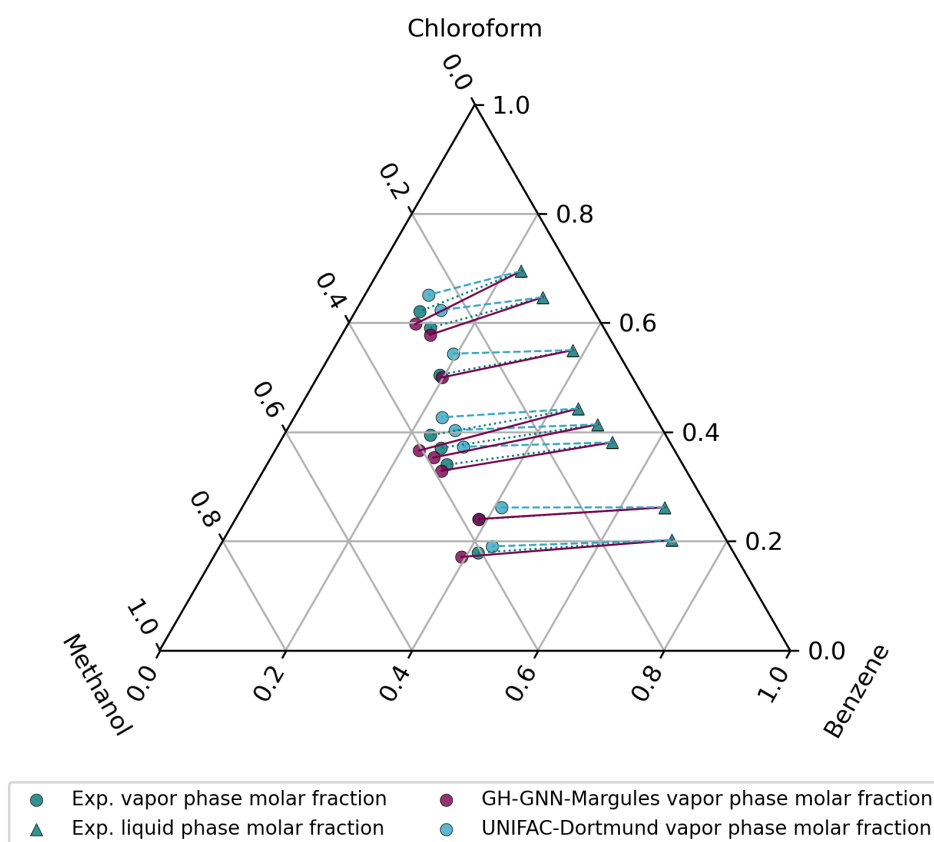


Fig. 5.9.: Ternary vapor-liquid equilibria for the system “chloroform/methanol/benzene” at 101.325 kPa. Experimental tie lines are taken from [92].

The performance results of GH-GNN-Margules for all ternary systems is shown in Table 5.6 grouped into isobaric and isothermal cases. A comparison to the UNIFAC-Dortmund model is also provided. Similarly to the case of binary systems presented in Section 5.2, UNIFAC-Dortmund is able to provide more accurate estimations of the ternary VLEs analyzed here. Despite this, the predictions of the GH-GNN-Margules model tend to also agree with the physical behavior of the mixtures. And, in fact, as with binary mixtures, the GH-GNN-Margules predictions of some systems align more closely to the experiments compared to the predictions of UNIFAC-Dortmund. Specifically, for the 10 ternary systems analyzed here, the GH-GNN-Margules model achieves a lower MAE than UNIFAC-Dortmund for the systems: “benzene/heptane/dimethylformamide” (MAE of 0.0296, compared to UNIFAC-Dortmund achieving 0.0297), “benzene/heptane/acetonitrile” (MAE of 0.0084,

Tab. 5.6.: GH-GNN-Margules and UNIFAC-Dortmund performance on predicting ternary vapor-liquid equilibria. The metrics are shown with respect to the predicted molar fractions of components 1 and 2 in the vapor phase.

	Isothermal			Isobaric		
	MAE ↓	R ² ↑	AE ≤ 0.03 ↑	MAE ↓	R ² ↑	AE ≤ 0.03 ↑
UNIFAC (Do)	0.0059	0.995	100%	0.0123	0.991	89.07%
GH-GNN-Margules	0.0568	0.394	37.96%	0.0330	0.925	66.60%

compared to UNIFAC-Dortmund with 0.0114) and “acetone/chloroform/benzene” (MAE of 0.0047, compared to 0.0184). For the other systems, the overall predictions of UNIFAC-Dortmund have lower MAE. However, even when the overall UNIFAC-Dortmund prediction of a specific system is better, at some conditions for that specific mixture, the predictions of the GH-GNN-Margules might outperform. An example of this is shown in Fig 5.9 for the system “chloroform/methanol/benzene” at 101.325 kPa.

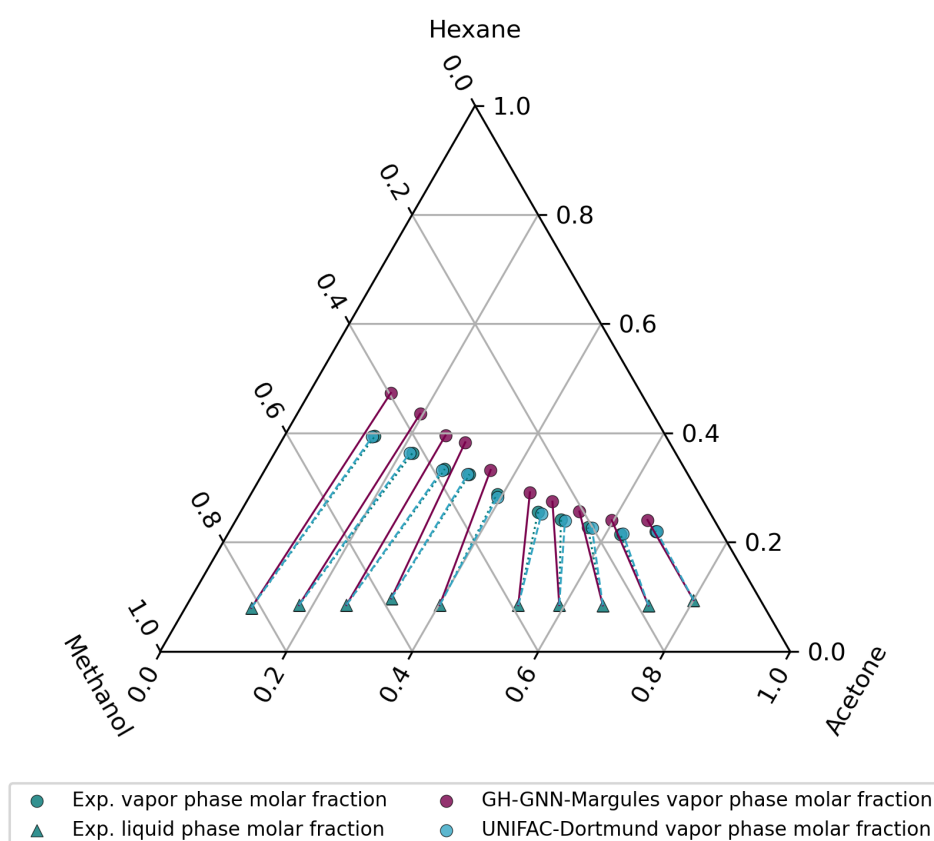


Fig. 5.10.: Ternary vapor-liquid equilibria for the system “hexane/methanol/acetone” at 313.15 K. Experimental tie lines are taken from [112].

For illustration, also the ternary diagram containing some points of the isothermal system analyzed here is shown in Fig. 5.10. In this case, the superior performance of UNIFAC-Dortmund is evident. Still, the general trend of the VLE is conserved by the GH-GNN-Margules. For systems, where UNIFAC-Dortmund is unable to provide predictions due to a lack of interaction parameters, the GH-GNN-Margules model can provide a first step approximation of the behavior that could be used in early stages of molecular and process design.

5.4 Chapter summary

This Chapter builds upon the framework established in Chapter 4, where the GH-GNN model was introduced, to now include the composition dependency of the activity coefficients. This extension is performed by further hybridization of the GH-GNN model with the extended Margules equation. The Margules equation is used in a sequential manner after the GH-GNN model, and predicts the most general activity coefficient γ_i exclusively from the predicted IDACs predicted by the GH-GNN. This means that the proposed framework, here referred to as the GH-GNN-Margules model, is capable of estimating activity coefficients by originally utilizing only available IDAC information for some limited systems during training.

The extended Margules equation approximates the molar excess Gibbs energy of the mixture using a Taylor polynomial. The expressions for the activity coefficients are then derived from the differential thermodynamic relationship of the molar excess Gibbs energy with respect to the number of moles of the corresponding chemical species. As a result, the resulting activity coefficients are thermodynamically consistent with the Gibbs-Duhem equation. Furthermore, the parameters of the Margules equation are expressed in terms of the IDACs of all binary mixtures that can be formed by the original mixture components, allowing the framework to depend solely on (scarce) experimental IDAC values.

The predictive performance of the proposed GH-GNN-Margules model is analyzed using an extensive collection of experimental binary vapor-liquid equilibria data. Isothermal and isobaric scenarios are considered, and the results show that the GH-GNN-Margules model is able to consistently predict many types of binary systems with remarkable accuracy. However, in this case, UNIFAC-Dortmund provides a more accurate estimation of the VLE behavior. This is directly attributed to the significant discrepancy in the available experimental data at the moment of parametrizing UNIFAC-Dortmund compared to the GH-GNN-Margules model. While

UNIFAC-Dortmund was parameterized using extensive IDAC, VLE, liquid-liquid equilibria (LLE), solid-liquid equilibria (SLE), azeotropic and caloric experimental measurements [59], the proposed GH-GNN-Margules model is able to predict VLEs solely from limited IDAC data. This disparity in data availability explains the accuracy gap, but it also shows the potential that GNN-based models might have if more experimental information is used during training.

A few ternary systems are also predicted with the proposed GH-GNN-Margules model, and compared to UNIFAC-Dortmund. Once again, UNIFAC-Dortmund outperforms, but exceptions on some systems can also be found despite the very limited number of systems considered and the popularity of the involved molecules. For those systems the proposed framework provides better estimates than UNIFAC-Dortmund. This also shows the potential that hybrid GNN-based models have in the context of predicting properties of multi-component mixtures from (limited) binary data alone. In a future step, the incorporation of various types of data (e.g., VLE, LLE, caloric) will be fundamental for the development of competitive and more efficient fluid-phase equilibria models by means of modern approaches like GNNs.

Applications to Separation Processes

” *...the machine at [Roger Sargent's] disposal had only 15 memories, each capable of storing 12 decimal digits...*

— **Costas Pantelides**

Professor of Chemical Engineering

In this Chapter, the GH-GNN model introduced in Chapter 4, and the GH-GNN-Margules model introduced in Chapter 5 are applied in the context of early stage separation process design. As discussed in previous Chapters, these models are able to estimate with remarkable accuracy the activity coefficients of a large set of components in liquid mixtures. This capability can be leveraged to assist the design of distinct separation processes. In specific, this Chapter focuses on the task of solvent screening for extractive separation systems. Moreover, the usefulness of GNN-based models is also exemplified in the context of biomass separation process design by estimating the solubility of biomass in different solvents.

6.1 Pre-selecting solvents for extractive distillation

Paradoxically, for some types of mixtures, the process of separation becomes easier if a new more “complex” mixture is created. This more “complex” mixture simply refers to a mixture with additional components. Here, a component (or multiple components) is added to the original mixture in order to modify the thermodynamic behavior of the system, and make the separation of the compounds of interest more practically viable.

A specific case of this type of processes occurs in extractive distillation systems. Let's assume that a liquid binary mixture of components i and j needs to be separated. While many separation routes can be possible, distillation is by far one of the most common choices for splitting mixtures in the chemical industry. However, if

the composition gap between the liquid and the vapor phases of the hypothetical mixture is very small, the separation through traditional distillation becomes very challenging. In fact, if the hypothetical mixture forms an azeotrope within the composition range of interest, the separation via traditional distillation becomes impossible [98]. The ease of separation through distillation of a mixture with key components i and j is given by the relative volatility

$$\alpha_{ij} = \frac{y_i/x_i}{y_j/x_j} = \frac{\gamma_i P_i^{sat}}{\gamma_j P_j^{sat}} \quad (6.1)$$

where, P_i^{sat} stands for the vapor pressure of pure component i , and similarly for component j . By introducing a solvent k , also called entrainer, the ratio of the activity coefficients of components i and j is modified. This ratio, in the presence of the entrainer, is called the selectivity

$$S_{ijk} = \left(\frac{\gamma_{ik}}{\gamma_{jk}} \right)_k \quad (6.2)$$

Therefore, different solvent candidates can be ranked according to their S_{ijk} value. The characteristics of the entrainer k should be such that when added to mixture i/j its relative volatility is increased by increasing the selectivity. It has to be also easy to recover in the downstream recovery distillation column, so that it can be recirculated back to the extraction column.

Energy efficiency is not the only important factor when selecting a good solvent. Many other aspects need to be considered within this complex task. For instance, the environmental impact of the new component k needs to be considered across many factors (e.g., toxicity, eutrophication, carcinogenicity, mutagenicity). Also, process safety considerations need to be taken into account (e.g., flammability, reactivity), along with socio-economic aspects (e.g., community impact).

Despite the high complexity of the solvent selection process, the increase in relative volatility induced by the candidate solvent is an important aspect that needs to be considered early on to rank solvent candidates, and to focus on the most promising candidates for further screening on other categories (e.g., environmental impact, cost, safety). For this reason, a pre-selection step based on relative volatility performance is relevant in early stages of separation process design.

Since for computing the selectivity values of different solvents only an estimation of the activity coefficients is needed, the previously proposed GH-GNN-Margules model can be effectively used for assisting this task. Moreover, if the selection process is

taken at infinite dilution conditions, the proposed GH-GNN model alone can be used for estimating the necessary IDACs.

In this Section, both models are used for the solvent pre-screening of three main types of challenging mixtures:

- **Aromatic/aliphatic:** represented by the binary mixtures of benzene with either hexane, heptane, octane, nonane or decane.
- **Paraffin/olefin/:** represented by the mixtures hexane/1-hexene, heptane/1-heptene and propane/propene.
- **Mixtures of oxygenated compounds:** represented by the mixtures of hexane with either methanol, ethanol, n-propanol, 2-propanol, acetone or 2-butanone.

The pre-selection calculations were performed considering an operating pressure of 101.325 kPa. The screening was conducted within the temperature range of the key component's normal boiling points using 5 evenly spaced values. The final ranking was obtained from the average performance among the 5 temperatures. Only solvents with a boiling point of at least 40 K higher than the highest boiling point of the two key components are considered to be feasible. The reason for this is that in the solvent regeneration step after the extraction column the process commonly becomes energetically attractive only if this difference in boiling points is present [97].

The pre-selection process was implemented considering a total of 700 candidate solvents. These 700 solvents were collected from [125] considering only organic molecules commonly used as solvents as indicated in [64]. The normal boiling point of each solvent was retrieved using the `Chemicals` library [14] giving preference to available experimental data. In case, the experimental normal boiling point was not available, an estimation was used from the Joback group contribution method [78]. The goal for this study is to identify whether the proposed pre-selection framework based on the proposed GNN-based models is able to produce solvent rankings that are consistent with the available information in the literature.

6.1.1 Selection based on selectivity at infinite dilution

The selectivity (Eq. 6.2) measures the relative impact that the addition of a solvent has on the non-ideal behavior of the two key components of the mixture. The maximum impact of this addition occurs when the key components are infinitely

diluted in the solvent. At this conditions, one can write the selectivity at infinite dilution as

$$S_{ijk}^{\infty} = \left(\frac{\gamma_{ik}^{\infty}}{\gamma_{jk}^{\infty}} \right)_k \quad (6.3)$$

where, γ_{ik}^{∞} denotes the IDAC of i infinitely diluted in k .

The selectivity S_{ijk}^{∞} is commonly used as a ranking criteria for solvents in extractive separation processes [25]. When the experimental IDAC values are available, one can use them in Eq. 6.3 to effectively perform a solvent pre-selection based on the expected modification of the mixture's key components relative volatility. However, as shown in Chapters 3 and 4, the available experimental IDAC data is very scarce when compared to the enormous chemical space of binary combinations that could be of potential interest. As a result, predictive methods can be used to estimate S_{ijk}^{∞} , allowing for the exploration of broader chemical and mixture spaces which might result in the discovery of better alternative solvents for extractive separation processes. Here, the GH-GNN model (introduced in Chapter 4) is used to estimate all the necessary IDAC values for screening over the 700 solvents mentioned before.

6.1.2 Selection based on relative volatility at infinite dilution

Screening solvents using the selectivity at infinite dilution (S_{ijk}^{∞}) ignores the relative effect of temperature changes in the relative volatility caused by the relative change on the key component's vapor pressure (cf. Eq. 6.4). When the range of temperatures considered during the screening is small, the contribution of the vapor pressures is small. In those cases, screening solvents using S_{ijk}^{∞} or α_{ijk}^{∞} becomes equivalent. However, if the temperature range during the screening is large, the relative change of the key component's vapor pressures can affect the order of the solvent ranking. For this reason, the pre-selection of the different mixtures considered here was also performed using the relative volatility at infinite dilution in the presence of the solvent α_{ijk}^{∞} , which can be written as

$$\alpha_{ijk}^{\infty} = \frac{\gamma_{ik}^{\infty} P_i^{sat}}{\gamma_{jk}^{\infty} P_j^{sat}} \quad (6.4)$$

and was compared to the solvent ranking produced by screening according to S_{ijk}^{∞} .

6.1.3 Selection based on minimum solvent-to-feed ratio

Both ranking metrics explained before (i.e., S_{ijk}^∞ and α_{ijk}^∞) consider the state of the system to be at infinite dilution in the solvent. However, such conditions are rarely representative of most of the stages in the extractive distillation columns. In fact, from an operational perspective, a small amount of solvent would be preferable. The reason for this is that the energetic requirements of the extraction and recovery columns decrease proportionally to the amount of solvent needed to carry out the desired thermodynamic change in the mixture. Therefore, another plausible and potentially more relevant criteria to rank solvents corresponds to the minimum amount of solvent needed to modify the relative volatility of the mixture to a desired threshold. This criteria is here referred to as the minimum solvent-to-feed ratio $\min SF$ to achieve a relative volatility value of 3.

The specific relative volatility threshold (i.e., 3) was identified by [25] to be a reasonable value for which the use of extractive distillation becomes attractive from an energetic point of view. The energetic requirements to carry out a separation via distillation decrease exponentially with the increase in relative volatility [20]. It has been reported that for relative volatility values greater than 3, the energy savings of increasing the relative volatility further are less significant compared to smaller values. Therefore, the same relative volatility threshold (i.e., $\alpha_{ijk} = 3$) is used here as the target value to calculate the $\min SF$.

Since the computation of α_{ijk} requires the estimation of the activity coefficients at finite compositions, the proposed GH-GNN-Margules model is employed here. For each solvent, temperature and set of compositions, an optimization problem needs to be solved in order to find the actual $\min SF$ that is required to achieve the desired change in the relative volatility. Here, the optimization problem is written as

$$\begin{aligned} \min SF = \operatorname{argmin}_{SF} \quad & (\alpha_{ijk}^- - \alpha_{ijk}^\wedge(SF))^2 \\ \text{s.t.} \quad & \alpha_{ijk}^\wedge(SF) = \frac{\gamma_{ik} P_i^{\text{sat}}}{\gamma_{jk} P_j^{\text{sat}}} \\ & \{\gamma_{ik}, \gamma_{jk}\} \leftarrow \text{GH-GNN-Margules}(SF) \\ & SF_{\min} \leq SF \leq SF_{\max} \end{aligned} \quad (6.5)$$

where, $\alpha_{ijk}^- = 3$ refers to the relative volatility threshold described before, and α_{ijk}^\wedge stands for the estimated relative volatility computed from the GH-GNN-Margules model and the vapor pressure KDB correlation (Eq. 5.9). The bounds SF_{\min} and SF_{\max} are specified here as 0 and 10,000, respectively.

Algorithm 3: Solvent pre-selection using the GH-GNN-Margules model and the minimum solvent-to-feed ratio criteria at standard pressure.

Data: List of solvents, vapor pressure parameters of key components i and j , normal boiling points $T_{b,i}$ and $T_{b,j}$.

Result: Ranking of solvents according to $\min SF$.

```

1 Initialize set of evenly distributed temperatures from  $\min(T_{b,i}, T_{b,j})$  to  $\max(T_{b,i}, T_{b,j})$  ;
  for solvent in list of solvents do
2   obtain  $K_1$  and  $K_2$  parameters for each binary pair of  $i, j$  and  $k \leftarrow$  GH-GNN model;
   for temperature in set of temperatures do
3      $P_i^{sat}, P_j^{sat} \leftarrow$  Eq. 5.9 ;
4      $\gamma_{ik}^\infty, \gamma_{jk}^\infty \leftarrow$  corresponding  $K_1$  and  $K_2$  parameters and Eq. 4.2 ;
5      $\alpha_{ijk}^\infty \leftarrow$  Eq. 6.4 ;
     if  $\alpha_{ijk}^\infty < 3$  then
6       solvent is unfeasible at this temperature;
     else
7       get  $w$  parameters for each binary pair of  $i, j$  and  $k \leftarrow$  GH-GNN model;
8       solve optimization problem 6.5 using the Brent's method;
     end
   end
9   get the average performance across temperatures ;
  end
10 rank solvents according to their  $\min SF$  value

```

Multiple component composition combinations can be obtained for a given solvent-to-feed ratio (SF). Here, 100 different combinations are computed for a given SF value using the following set of equations

$$\mathbf{x}_k = \left[\frac{SF}{1 + SF} \right]_{a=1}^{100} \quad (6.6)$$

$$\mathbf{x}_i = \left[\frac{1 - SF/(1 + SF)}{100} \cdot (a - 1) \right]_{a=1}^{100} \quad (6.7)$$

$$\mathbf{x}_j = 1 - \mathbf{x}_k - \mathbf{x}_i \quad (6.8)$$

where, \mathbf{x}_i , \mathbf{x}_j and \mathbf{x}_k are the ordered arrays of composition values for key components i, j and solvent k , respectively. The estimated value α_{ijk}^∞ in the optimization problem 6.5 is taken as the minimum α_{ijk}^∞ value across all the 100 composition combinations for the given SF . This is to ensure that the relative volatility threshold is met across the entire range of possible compositions.

It is important to mention that for the solvent selection process carried out here, a total of 21,695 distinct IDAC values are needed. Out of these, only 260 ($\sim 1.2\%$) were actually observed by the GH-GNN model during training. This, once

again, shows the scarcity of available IDAC experimental data. As a result, the GH-GNN model needs to either interpolate or extrapolate to other chemical species in order to predict the necessary IDAC values. For comparison, at least 1,248 ($\sim 5.7\%$) IDACs would be unfeasible to predict using UNIFAC-Dortmund due to an unfeasible fragmentation of the molecules involved. This number is, however, expected to increase when considering cases where the fragmentation is possible, but the necessary binary interaction parameters are not available. Therefore, while the solvent screening using UNIFAC-Dortmund cannot be performed for the entirety of the 700 solvents considered here, the GH-GNN model is able to screen across all solvents. The screening of solvents using experimental information, is of course much more restricted. This shows the advantage of the proposed GNN-based framework for exploring more extensive chemical spaces in early stages of separation process design.

Algorithm 3 shows the solvent pre-selection procedure used in this work according to the $\min SF$ criteria. The optimization problem in line 8 is solved using the `SciPy` library [158] with the Brent's method using maximum 2000 iterations and a tolerance of 1.48×10^{-8} . A Jaccard distance threshold of 0.7 (as introduced in Section 4.1.5) was considered here for selecting solvents based on results that are expected to be accurate, while still allowing for an extensive exploration of the chemical space. Only 3 solvents would have been selected as part of the top candidates that have a Jaccard distance metric above this threshold, and that are here discarded.

6.1.4 Results of solvent pre-selection

Aromatic/aliphatic mixtures

Table 6.1 shows the top 5 solvents selected for the considered aromatic/aliphatic mixtures according to the 3 criteria introduced before (i.e., selectivity at infinite dilution S_{ijk}^∞ , relative volatility at infinite dilution α_{ijk}^∞ and minimum solvent-to-feed ratio to achieve a relative volatility of 3 $\min SF$). By looking at the ranking of, for instance the benzene/octane mixture, it can be observed that the selection obtained from the S_{ijk}^∞ and α_{ijk}^∞ metrics is for the most part consistent, which reinforces the fact that the influence of the vapor pressure across relatively small temperature ranges is minimal. However, the solvent ranked at position 5 differs between both metrics, which also shows that the vapor pressure estimation can sometimes change the exact solvent performance evaluation relative to others.

Tab. 6.1.: Top 5 solvents selected for the indicated aromatic/aliphatic mixtures. The number between parenthesis indicates the value for the corresponding metric.

Rank	Selection criteria		
	S_{ijk}^{∞}	α_{ijk}^{∞}	minSF
Benzene/hexane			
1	succinonitrile (16.7)	succinonitrile (23.9)	dimethyl sulfoxide (0.4)
2	formamide (16.6)	formamide (23.6)	tetramethylene sulfoxide (0.5)
3	ethanolamine (13.4)	ethanolamine (19.2)	tetramethylurea (0.6)
4	sulfolane (13.2)	sulfolane (18.9)	n,n-dimethyl acetamide (0.6)
5	tetramethylene sulfone (13.2)	tetramethylene sulfone (18.9)	3,4-dimethyl phenol (0.6)
Benzene/heptane			
1	ethylene glycol (28.4)	ethylene glycol (16.4)	dimethyl sulfoxide (1.8)
2	formamide (25.3)	formamide (14.6)	ethylene glycol (1.9)
3	dimethyl sulfone (20.4)	dimethyl sulfone (11.8)	formamide (2.6)
4	glycerol (18.2)	glycerol (10.5)	succinonitrile (2.8)
5	succinonitrile (15.2)	succinonitrile (8.8)	glycerol (2.8)
Benzene/octane			
1	formamide (46.2)	formamide (12.0)	formamide (4.4)
2	ethylene glycol (30.3)	ethylene glycol (7.9)	ethylene glycol (4.9)
3	dimethyl sulfone (25.0)	dimethyl sulfone (6.6)	glycerol (6.7)
4	glycerol (24.5)	glycerol (6.3)	dimethyl sulfone (7.0)
5	sulfolane (20.9)	tetramethylene sulfone (5.4)	sulfolane (8.5)
Benzene/nonane			
1	formamide (73.1)	formamide (9.4)	formamide (6.5)
2	ethylene glycol (39.4)	ethylene glycol (5.1)	ethylene glycol (12.4)
3	glycerol (33.6)	glycerol (4.2)	glycerol (21.8)
4	sulfolane (28.5)	dimethyl sulfone (3.6)	tetramethylene sulfone (34.1)
5	tetramethylene sulfone (28.5)	tetramethylene sulfone (3.6)	sulfolane (34.1)
Benzene/decane			
1	formamide (108.5)	formamide (7.5)	formamide (10.6)
2	glycerol (65.2)	glycerol (4.4)	glycerol (27.2)
3	succinonitrile (38.1)	succinonitrile (2.3)	
4	malononitrile (33.5)	malononitrile (2.2)	
5	sulfolane (31.3)	dimethyl sulfone (2.1)	

In general, the selection of the best solvents is consistent with the information that can be found in the literature. For instance, sulfolane has been already reported as a suitable entrainer for aromatic/aliphatic separations [154, 48]. Similarly, for formamide and succinonitrile which have been reported for a relatively long time as suitable solvents for extractive distillation of aromatic/aliphatic mixtures [16, 142]. Ethylene glycol has been also repeatedly selected as a promising solvent by the proposed framework, which is in accordance of the reported patent of [29] that suggests several glycols as potential solvents for aromatic/aliphatic separations. These results suggest that the proposed methodology (based on the GNN-based models for predicting activity coefficients) is able to rank solvents for extractive distillation in a consistent manner when compared to the literature.

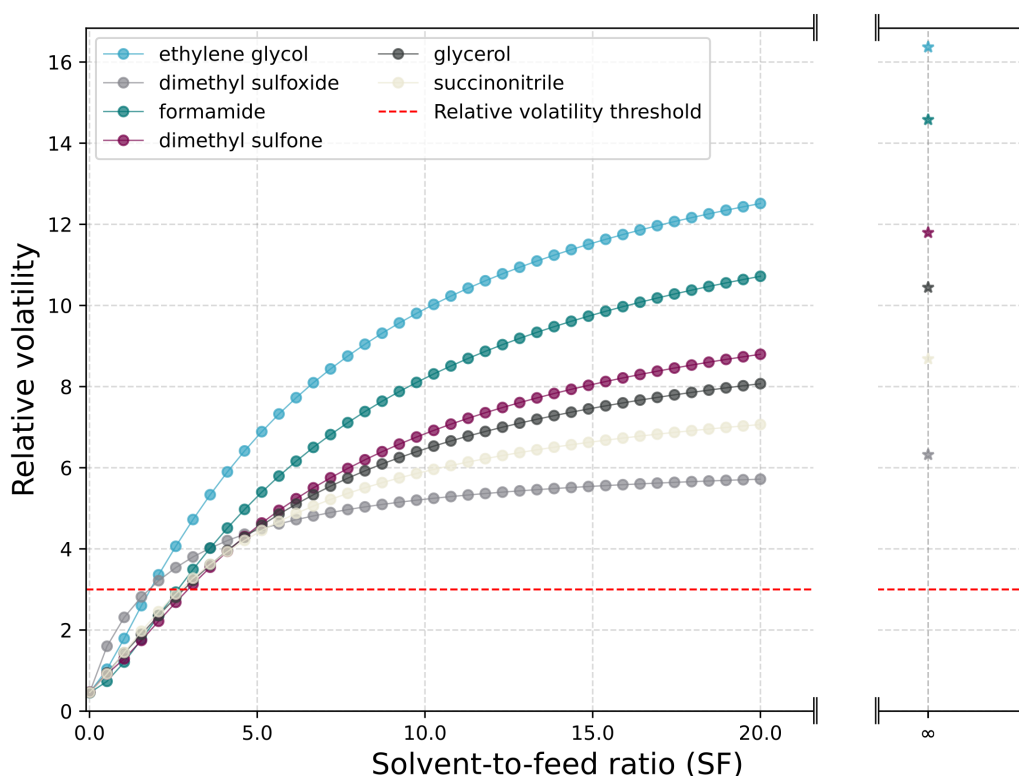


Fig. 6.1.: Effect on the relative volatility of the mixture benzene/heptane caused by different solvent-to-feed ratios (SF) of the top 5 solvents identified in the pre-selection process.

The discrepancy between the rankings obtained by the metrics at infinite dilution (i.e., S_{ijk}^{∞} and α_{ijk}^{∞}) and the one at finite concentrations (i.e., $\min SF$) is caused by the different rate at which the amount of solvent affects the relative volatility of the system. Fig. 6.1 exemplifies the different behavior of solvents in the mixture benzene/heptane according to their relative amount in the mixture. First, it can

be observed, as expected, that the maximum increase in relative volatility occurs at infinite dilution (i.e., star markers in Fig. 6.1). The change in relative volatility decreases with the decrease of the solvent amount in the mixture, as reflected by the different solvent-to-feed ratios (SF). It can be observed that, if the selection of the solvents is carried out based on the relative volatility at infinite dilution (α_{ijk}^∞), ethylene glycol appears to be better than dimethyl sulfoxide by a significant margin. However, when selecting solvents based on $\min SF$ it is dimethyl sulfoxide the one outperforming the rest. This is reflected by the fact that the mixture's relative volatility trajectory in the presence of dimethyl sulfoxide crosses the threshold (i.e., $\alpha_{ijk} = 3$) at a relatively smaller SF compared to the other solvents. In general, this shows that a solvent's impact on a given mixture's relative volatility can decay more rapidly than that of other solvents due to the complex interactions at different concentrations. Hence, the selection of solvents based on metrics at infinite or finite concentrations can differ from each other.

Nevertheless, what is important to highlight is the significant difference in the relative ranking that the metrics S_{ijk}^∞ and α_{ijk}^∞ have when compared to the $\min SF$ criteria. For instance, for the benzene/hexane case, while the first two metrics did not classified dimethyl sulfoxide within the top 5 solvents, its performance according to $\min SF$ suggests that is the most promising solvent to carry out the separation. The suitability of dimethyl sulfoxide is confirmed by the patent [48], along with the candidate *n,n*-dimethyl acetamide, which the first two metrics also did not classified in the top 5. Dimethyl phenol has also been reported as promising solvent [15]. For other mixtures, the solvent selection is more consistent across all selection criteria. For example, by suggesting formamide as the most suitable solvent for the mixtures of benzene with octane, nonane and decane.

It is also interesting to note that, as indicated in Table 6.1, only 2 solvents were found to increase the relative volatility of the benzene/decane mixture above the specified threshold. The solvents succinonitrile, malonitrile and dimethyl sulfone are not able to increase the relative volatility of the mixture to the desired level even at infinite dilution conditions. Since, the conditions at finite concentrations more closely follow the actual state of extractive distillation columns, the $\min SF$ is here recommended compared to the other two criteria. Even when the computation of $\min SF$ is computationally more intensive than S_{ijk}^∞ or α_{ijk}^∞ , the difference is not significant given the proposed framework. Specifically, while the screening across the 700 candidate solvents considered here takes around 2 minutes for S_{ijk}^∞ and α_{ijk}^∞ , the screening using $\min SF$ takes 6 minutes. This times are accomplished in an AMD Ryzen 9 6900HS with 16.0 GB and a RTX GPU.

Paraffin/olefin mixtures

Table 6.2 shows the top 5 solvents selected for each of the paraffin/olefin mixtures considered here. In this case, the selection of the top 5 solvents according to S_{ijk}^{∞} and α_{ijk}^{∞} correspond exactly to each other. For the mixtures hexane/1-hexene and heptane/1-heptene no solvent was found to increase the relative volatility to at least 3. This is expected given the difficulty of separating two compounds that only differ by an unsaturated bond. Solvents that have been reported for the separation of paraffin/olefin mixtures were selected by the framework, including dimethyl sulfoxide [31], dimethyl formamide [141], acetonitrile [99] and similar compounds like n,n-dimethyl acetamide [15]. The appearance of 2,6-dimethyl phenol among the top solvents for separating heptane from 1-heptene aligns with the patent of [17] reporting the use of a phenol-derived species for the same purpose. Moreover, hydrogen-bond donor species, like glycerol and ethylene glycol, have been selected by the proposed framework, which is supported by the use of similar species in the context of deep eutectic solvents (DES) for extraction of paraffin/olefin mixtures [62].

Tab. 6.2.: Top 5 solvents selected for the indicated paraffin/olefin mixtures. The number between parenthesis indicates the value for the corresponding metric.

Rank	S_{ijk}^{∞}	Selection criteria	
		α_{ijk}^{∞}	minSF
hexane/1-hexene			
1	succinonitrile (3.3)	succinonitrile (2.8)	
2	n,n-dimethyl acetamide (2.7)	n,n-dimethyl acetamide (2.3)	
3	quinoline (2.5)	quinoline (2.1)	
4	2-chlorocyclohexanone (2.5)	2-chlorocyclohexanone (2.1)	
5	diketene (2.3)	diketene (1.9)	
heptane/1-heptene			
1	ethylene glycol (3.2)	ethylene glycol (2.8)	
2	2-chlorocyclohexanone (3.2)	2-chlorocyclohexanone (2.7)	
3	quinoline (2.8)	quinoline (2.4)	
4	2,6-dimethyl phenol (2.5)	2,6-dimethyl phenol (2.1)	
5	dimethyl formamide (2.4)	dimethyl formamide (2.0)	
propane/propene			
1	3-bromopropyne (6.9)	3-bromopropyne (5.4)	glycerol (2.7)
2	glycerol (6.8)	glycerol (5.3)	dimethyl sulfoxide (3.4)
3	1,4-dichloro-2-butyne (6.2)	1,4-dichloro-2-butyne (4.8)	3-bromopropyne (6.6)
4	3-chloropropyne (6.2)	3-chloropropyne (4.8)	acetonitrile (6.9)
5	1,1,2-trichloroethane (5.8)	1,1,2-trichloroethane (4.5)	1,4-dichloro-2-butyne (7.2)

Mixtures of oxygenated compounds

Similarly to the previous mixture types, Table 6.3 shows the top 5 solvents selected for the mixtures considered here that include oxygenated compounds. The consis-

tency of the selection between the S_{ijk}^{∞} and α_{ijk}^{∞} metrics is also clear, corresponding exactly to each other in the ranking of the top 5 candidates. In general glycol species (e.g., ethylene glycol, diethylene glycol, propylene glycol and triethylene glycol) are consistently selected among the most promising solvents for the deoxygenation of hexane mixtures. All of these species have high affinity for the oxygenated compound (due to hydrogen-bonding), which causes them to entrain it in the mixture facilitating the separation.

Tab. 6.3.: Top 5 solvents selected for the indicated mixtures containing oxygenated compounds. The number between parenthesis indicates the value for the corresponding metric.

Rank	Selection criteria		
	S_{ijk}^{∞}	α_{ijk}^{∞}	minSF
hexane/methanol			
1	formamide (411.4)	formamide (356.1)	ethylene glycol (0.9)
2	ethylene glycol (255.3)	ethylene glycol (221.0)	triethylene glycol (1.0)
3	ethanolamine (171.0)	ethanolamine (148.1)	formamide (1.0)
4	glycerol (159.1)	glycerol (137.7)	triethanolamine (1.0)
5	diethanolamine (104.6)	diethanolamine (90.5)	diethylene glycol (1.0)
hexane/ethanol			
1	formamide (256.6)	formamide (364.4)	triethylene glycol (0.6)
2	ethylene glycol (157.7)	ethylene glycol (224.1)	diethylene glycol (0.7)
3	ethanolamine (115.9)	ethanolamine (164.8)	triethanolamine (0.7)
4	glycerol (77.0)	glycerol (109.4)	ethylene glycol (0.7)
5	diethanolamine (75.0)	diethanolamine (106.6)	1,4-butanediol (0.7)
hexane/n-propanol			
1	formamide (120.1)	formamide (334.4)	triethylene glycol (0.4)
2	ethylene glycol (97.5)	ethylene glycol (272.0)	1,4-butanediol (0.4)
3	ethanolamine (77.8)	ethanolamine (217.7)	diethylene glycol (0.4)
4	1,3-butanediol (46.2)	1,3-butanediol (128.4)	triethanolamine (0.5)
5	dimethyl sulfoxide (44.1)	dimethyl sulfoxide (124.2)	diethanolamine (0.5)
hexane/2-propanol			
1	formamide (173.7)	formamide (286.0)	triethylene glycol (0.5)
2	ethylene glycol (110.4)	ethylene glycol (181.8)	diethylene glycol (0.5)
3	ethanolamine (97.8)	ethanolamine (161.3)	1,4-butanediol (0.6)
4	glycerol (69.3)	glycerol (114.1)	triethanolamine (0.6)
5	propylene glycol (53.7)	propylene glycol (88.5)	diethanolamine (0.6)
hexane/acetone			
1	formamide (191.0)	formamide (127.6)	triethylene glycol (0.8)
2	glycerol (78.4)	glycerol (52.4)	diethylene glycol (0.8)
3	succinonitrile (73.5)	succinonitrile (49.1)	formamide (0.9)
4	ethanolamine (65.0)	ethanolamine (43.4)	triethanolamine (0.9)
5	ethylene glycol (55.4)	ethylene glycol (37.0)	diethylenetriamine (0.9)
hexane/2-butanone			
1	formamide (92.4)	formamide (130.2)	triethylene glycol (0.4)
2	glycerol (37.6)	glycerol (53.0)	diethylene glycol (0.5)
3	succinonitrile (33.7)	succinonitrile (47.5)	triethanolamine (0.5)
4	ethylene glycol (32.1)	ethylene glycol (45.3)	1,4-butanediol (0.5)
5	ethanolamine (28.7)	ethanolamine (40.4)	diethylenetriamine (0.5)

Dimethyl sulfoxide has been suggested as a promising candidate for the hexane/n-propanol mixture according to the S_{ijk}^{∞} and α_{ijk}^{∞} metrics. This aligns with the patent

of [135]. Furthermore, the selection of butanediol species and glycerol also aligns with the results reported in the literature [172] where solvent design (instead of solvent selection) was used. This further strengthens the value and effectiveness of the proposed solvent pre-selection framework that utilizes GNN-based models as predictive tools during the screening process.

6.2 Pre-selecting solvents for liquid-liquid extraction of caprolactam from ionic-liquid

This Section provides an additional example of how the proposed GNN-based framework can be used for effectively pre-selecting candidate solvents for a liquid-liquid extraction process to separate caprolactam from the ionic liquid ethyl-3-methylimidazolium tetrafluoroborate ([EMIM][BF₄]). This specific case-study was selected based on a collaborative effort with Ann-Joelle Minor. My sole contribution consisted in the solvent pre-selection process using the proposed GNN-based models, and it is part of the work shown here.

The importance of separating caprolactam from [EMIM][BF₄] is evident in the context of chemical recycling of plastics, specifically the chemical recycling of polyamide 6. In this process, ionic liquids have been recently investigated [79] as promising candidates to carry out the solvolytic depolymerization of polyamide 6 into caprolactam. The resulting stream contains mainly caprolactam and the ionic liquid, which has to be separated into its (close to) pure components to be able to recycle the ionic liquid back to the process and to re-utilize the caprolactam in a circular economy fashion.

The solvent pre-selection was performed using the same 700 solvents considered in Section 6.1. However, instead of considering temperatures between the boiling points of the two key components (as in the previous Section dealing with extractive distillation), here the temperature range for the solvent selection was selected as 298.15 to 313.15 K. This temperature range is chosen so as to approximate the actual process conditions of the liquid-liquid extraction unit. As with the extractive distillation process, 5 evenly spaced temperatures were considered within this range, and the final solvent performance corresponds to the mean performance over these 5 temperatures.

For liquid-liquid extraction processes it is common to define a metric for the solvent performance in the separation, this metric is usually refer to as the separation factor K_α , which is defined as

$$K_\alpha = \frac{\gamma_{CL}^R \gamma_{IL}^E}{\gamma_{CL}^E \gamma_{IL}^R} \quad (6.9)$$

where, the superscripts R and E refer to the raffinate and extract phases, respectively. The subscripts CL and IL refer to caprolactam and [EMIM][BF4], respectively. Eq. 6.9 not only considers the effectiveness of the solvent at the extraction phase (measured by the ratio $\gamma_{CL}^R/\gamma_{CL}^E$), but also at the solvent recovery phase (measured by the ratio $\gamma_{IL}^E/\gamma_{IL}^R$).

In order to consider the solvent performance at both the extraction and recovery phases, the solvent pre-selection was carried out by defining the following separation factor at infinite dilution

$$K_\infty = \frac{\gamma_{S,IL}^\infty}{\gamma_{CL,S}^\infty} \quad (6.10)$$

where, the subscript S refers to the solvent. In this Equation, $\gamma_{S,IL}^\infty$ approximates the performance of the solvent at the recovery phase, where it has to be separated from the extract phase that is rich in [EMIM][BF4]. Similarly, $\gamma_{CL,S}^\infty$ measures the solvent performance at the extraction phase, in which caprolactam has to be separated from the solvent-rich mixture.

Therefore, for computing the necessary IDACs to calculate K_∞ , the extended GH-GNN model trained simultaneously for predicting organic and ionic liquid solvents (as presented in Section 4.2) was used. Additionally, the following filters were applied to the candidate solvents: the solvent's normal boiling point should be less than or equal to 473.15 K, its molecular weight should be less than or equal to 135 g/mol and its Globally Harmonized System (GHS) signal classification should not be "danger" or unclassified. The boiling point restriction is in place to facilitate the downstream solvent's recovery phase from the ionic-liquid-rich extract. The molecular weight constraint is implemented to manage the solvent's viscosity, which is crucial for operational efficiency. Additionally, the GHS constraint guides the selection of solvents that pose potentially lower risks to environmental, health and safety considerations.

Table 6.4 shows the top 10 solvents to perform the separation according to the performance metric given in Eq. 6.10, and after applying the previously described

Tab. 6.4.: Top 10 solvents selected for the liquid-liquid extraction of caprolactam from [EMIM][BF₄] after considering the normal boiling point, molecular weight and GHS classification filters. The ranking metric used corresponds to Eq. 6.10. The conventional solvent is marked with *.

Rank	Solvent	K_{∞}	Jaccard distance metric
1	2-Octanol	11.945	0.122
2	6-Methyl-1-heptanol	11.783	0.286
3	2-Ethylhexan-1-ol	10.632	0.419
4	1-Octanol	9.702	0
5	Butylcyclopentane	7.677	0.149
6	Isobutyl acrylate	6.256	0.451
7	Butyl acrylate	6.190	0.362
8	1-Butoxy-2-propanol	6.176	0.390
9	Isoamyl acetate	5.822	0.256
10	Dibutyl ether	5.600	0.235
431	Ethyl acetate*	0.627	0.317

filters. Moreover, the performance of the conventional solvent (ethyl acetate) [79] for this process is shown for comparison. The Jaccard distance metric (as given in Section 4.1.5) is also shown for each solvent.

It is worth mentioning that for the case of ethyl acetate, its GHS signal classification is “danger”. Hence, this solvent would be discarded by the proposed set of filters. However, even if the GHS criteria is ignored, the performance of ethyl acetate is predicted to be significantly less promising than that of the other candidate solvents. Various alcohol species appear among the most promising candidates. Specifically, 1-octanol and 2-octanol stand out by their low Jaccard distance metric, which, as shown in Section 4.1.5, inversely correlates with the accuracy of the model. Among these two, 1-octanol is available at a cheaper price compared to 2-octanol [105].

Therefore, in this collaborative work [105], the phase equilibrium of the proposed 1-octanol candidate solvent in the presence of caprolactam and [EMIM][BF₄] was analyzed experimentally. This is shown in Fig. 6.2. A comparison with the LLE predictions of COSMO-RS (with BP86-TZVPD-FINE level of theory) is also available. From the experiments it can be observed that at low concentrations of the solvent, the system in the presence of 1-octanol contains solids at equilibrium. More importantly, the predictions of COSMO-RS of the system in the presence of 1-octanol agree with the experiments, and show that the miscibility gap is indeed larger than that of the system in the presence of the conventional solvent (i.e., ethyl acetate). These experimental observations agree with the solvent selection performed using the extended GH-GNN model.

In order to compare the ranking of solvents as predicted by the GH-GNN model, the UNIFAC-IL [32] model was also used to compute the metric in Eq. 6.10 and to rank

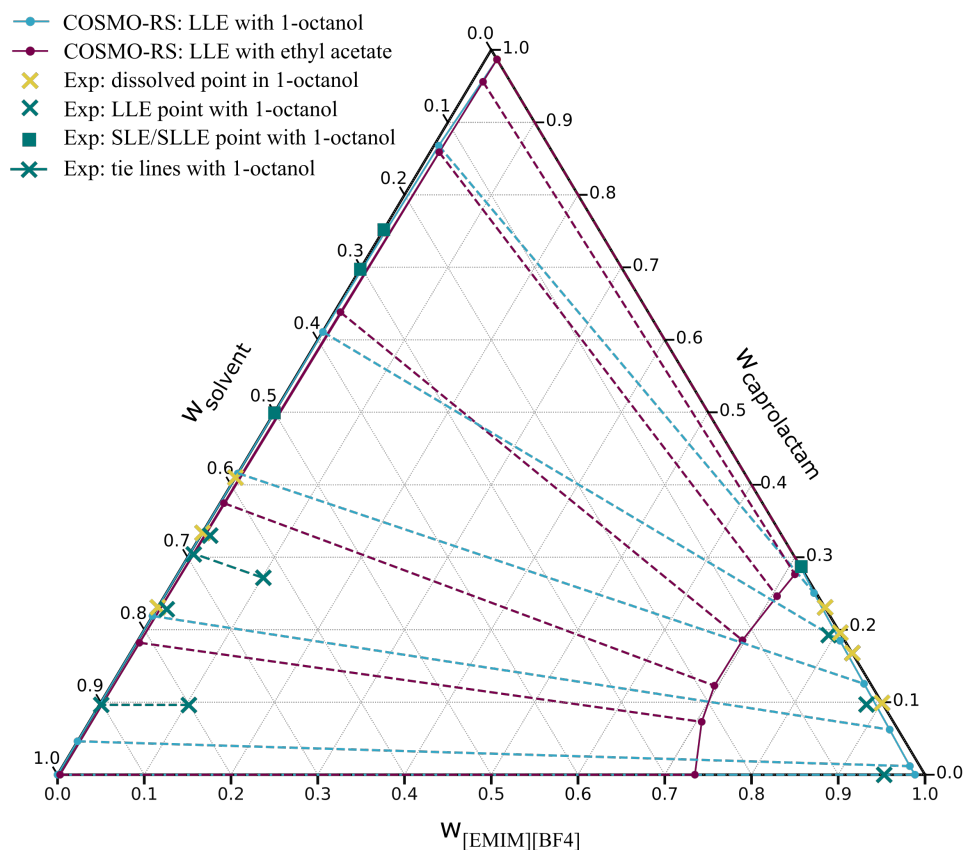


Fig. 6.2.: Phase equilibrium behavior of the ternary system $[EMIM][BF_4]$, caprolactam and the indicated solvent as predicted by COSMO-RS, and measured experimentally. Adapted from [105].

the solvents accordingly. However, since UNIFAC-IL is restricted by the feasibility of molecule fragmentation and availability of binary interaction parameters, the necessary IDACs for only 349 (50%) out of the 700 solvents could be predicted by UNIFAC-IL. These are the solvents included in this comparison.

Fig. 6.3 shows the comparison between the rankings obtained by these two methods. While significant discrepancies in the ranking of the solvents can be observed when comparing both models, a correlation exist between the rankings of both methods. In this case, the Pearson's correlation coefficient is 0.62. Since the accuracy of the extended GH-GNN model on predicting IDACs tends to be higher than that of UNIFAC-IL (cf. Section 4.2), it is expected that the solvent ranking produced by the GH-GNN model is more accurate. Moreover, as pointed out earlier, the space of the solvents that UNIFAC-IL can consider is more limited than that of the GH-GNN model. These two are clear advantages of the proposed model for performing solvent pre-selection when compare to UNIFAC-based methods.

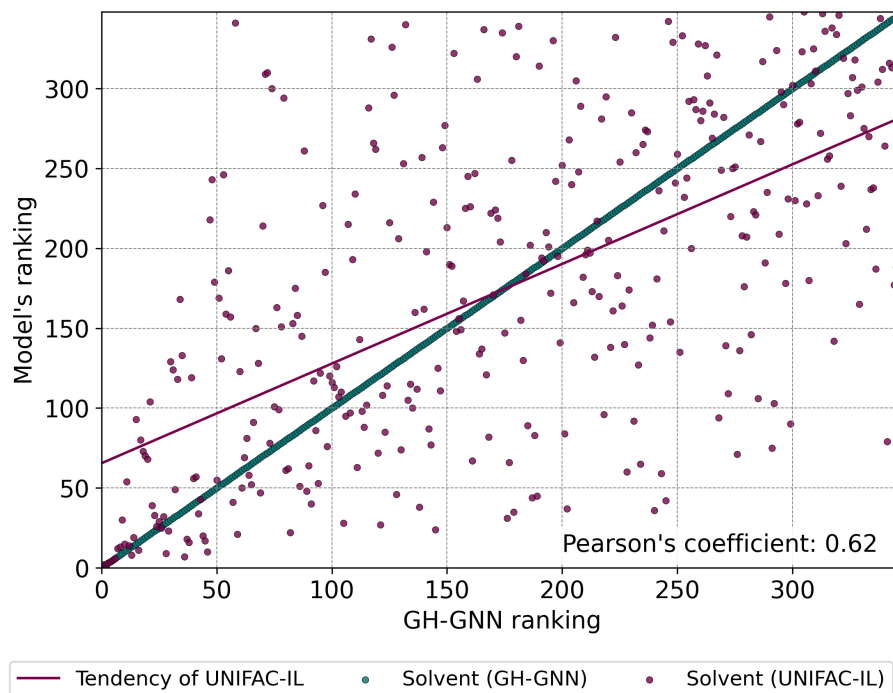


Fig. 6.3.: Comparison of solvent rankings according to the extended GH-GNN model and UNIFAC-IL [32] for the extraction case-study of [EMIM][BF₄] and caprolactam.

The replacement of the conventional solvent by 1-octanol not only increases the modeled process efficiency, but also its overall profitability [105]. The effectiveness of alcohol species to extract caprolactam (albeit not necessarily in the presence of ionic liquids) has also been reported in the literature [134, 168], including examples such as 1-octanol and 2-ethylhexan-1-ol, which are also selected by the extended GH-GNN model. This suggests that, also for this case-study with presence of an ionic-liquid, the obtained solvent ranking is able to suggest promising candidates effectively. Therefore, this case-study exemplifies once again the utility of the proposed hybrid GNN-based models in supporting the engineer or researcher in the early stages of chemical process design.

6.3 Graph neural networks assisting the design of a lignin fractionation process

This Section provides yet another example of how GNN-based models can be used for supporting the design of separation processes. Specifically, this Section focus on

supporting the design of lignin bio-refineries. However, distinct to the previous two case-studies focusing on solvent screening for extractive distillation and liquid-liquid extraction, the GNN model here is trained to predict the solubility of lignin in distinct organic solvents. The work presented in this Section represents my contribution to a collaborative effort led by Laura König-Mattern [88].

Distinct to phase equilibria problems, which were the main focus of this dissertation, solubility refers to the ability of a solvent species to form a single homogeneous mixture with the solute. Estimating the solubility of lignin in various solvents is relevant for optimizing the so-called organosolv process, which is a promising path towards the development of bio-refineries. The solvent selection process can be carried out in the form of a solvent screening (as exemplified for other systems in Sections 6.1 and 6.2), or in the form of a solvent design problem. In both cases, a predictive model that is accurate for predicting the solubility of lignin is desired.

However, a significant challenge for developing predictive models for lignin solubility is the scarcity of experimental data. For this reason, previous approaches have relied on the predictions of COSMO-RS using a representative molecule of lignin consisting in a trimer of guaiacyl connected with β -O-4 bonds [87]. This representative molecule was used by my collaborator to compute the solubility in 3,314 distinct solvents using COSMO-RS (with BP86-TZVPD-FINE level of theory) at 343.15 K.

Despite being a sole approximation of the lignin solubility, the resulting data set could be used to rank the performance of the solvents considered. However, if an additional solvent is to be screened, the corresponding quantum mechanical and COSMO-RS calculations should be performed, which could be a considerably time-consuming process. If the quantum mechanical calculations are not yet available for the solvent of interest, such computation may span from one hour up to a week on a typical CPU machine. This is an important limitation if a large chemical space is desired to be explored.

With this motivation, a GNN model was trained to predict the solubility of lignin in various solvents as predicted by COSMO-RS. In this case, since the accuracy of the GNN model is bounded by the accuracy of the COSMO-RS predictions, the GNN model acts as a plain surrogate to speed-up the solubility estimation process. For developing the GNN, the data split was performed using the so-called Butina clustering algorithm as implemented in `RDKit` [127] (version 2021.03.1). This algorithm clusters the molecules into similar groups according to the Jaccard distance of their fingerprints. In this case, the Morgan fingerprint was used with a radius of 2 and a size of 2048 bits. Then, each of the resulting clusters was split randomly with an 80/20 proportion to define the train and test sets. In case a

Tab. 6.5.: Atomic features defining the initial feature vector of nodes in the molecular graphs constructed from the lignin solubility COSMO-RS calculations. The dimension of the corresponding one-hot encoded feature is also shown.

Feature	Description	Dimension
Atom type	[C, O, N, Cl, F, S, Si, Br, P, Se, I, B, As, Ge, Al]	15
Ring	Is the atom in a ring?	1
Aromatic	Is the atom part of an aromatic system?	1
Hybridization	[sp, sp ² , sp ³ , sp ³ d]	4
Bonds	Number of bonds the atom is involved in. [0,1,2,3,4]	5
Charge	Atom’s formal charge. [0, -1, 1, 3]	4
Attached Hs	Number of bonded hydrogen atoms. [0,1,2,3]	4

Tab. 6.6.: Bond features defining the initial feature vector of edges in the molecular graphs constructed from the lignin solubility COSMO-RS calculations. The dimension of the corresponding one-hot encoded feature is also shown.

Feature	Description	Dimension
Bond type	[Single, double, triple, aromatic]	4
Conjugated	Whether the bond is conjugated	1
Ring	Whether the bond is part of a ring	1

cluster consists only of 4 or less molecules, all of these were placed in the training set automatically. Out of the training set, 15% of the data was randomly selected and used for model validation.

The atomic and node features used for defining the solvent molecular graphs are given in Tables 6.5 and 6.6, respectively. The GNN model consists of 3 message-passing layers with a node embedding-size of 50. The GNN architecture is similar to the one used for developing the GNN model for predicting isothermal IDACs introduced in Section 3.3. The initial set of node features is transformed using Eq. 3.1, followed by the 3 message-passing layers computed by Eq. 3.2. The activation function in the last message-passing layer was removed. The edge-transforming function consisted of a single hidden-layer neural network of size 64 with the ReLU activation. Batch normalization was also used according to Eq. 3.3 after each message-passing layer. The final vectorial representation of the solvent was computed using the global max operator that can be written as:

$$\mathbf{a}_g = \text{Max}(\{\mathbf{a}_v^{final} | v \in \mathcal{V}\}) \quad (6.11)$$

where the Max operation is applied element-wise across the set of node embeddings of the graph.

The resulting graph embedding \mathbf{a}_g is then fed to a multi-layer perceptron consisting of 2-hidden-layers with dimensions 50 and 25. Dropout was used across the model with a probability of 0.1. This GNN model was trained over 100 epochs with the AdamW optimizer and with a learning rate of 0.001 using batches of 32 solvents to minimize the MSE function. A linear learning rate scheduler was used with a factor of 0.8 and a patience of 3 epochs.

The final model consist of an ensemble of 5 GNNs trained in the same fashion but with different train/validation splits. The final prediction of the ensemble model was obtained by averaging the predictions of these 5 GNNs.

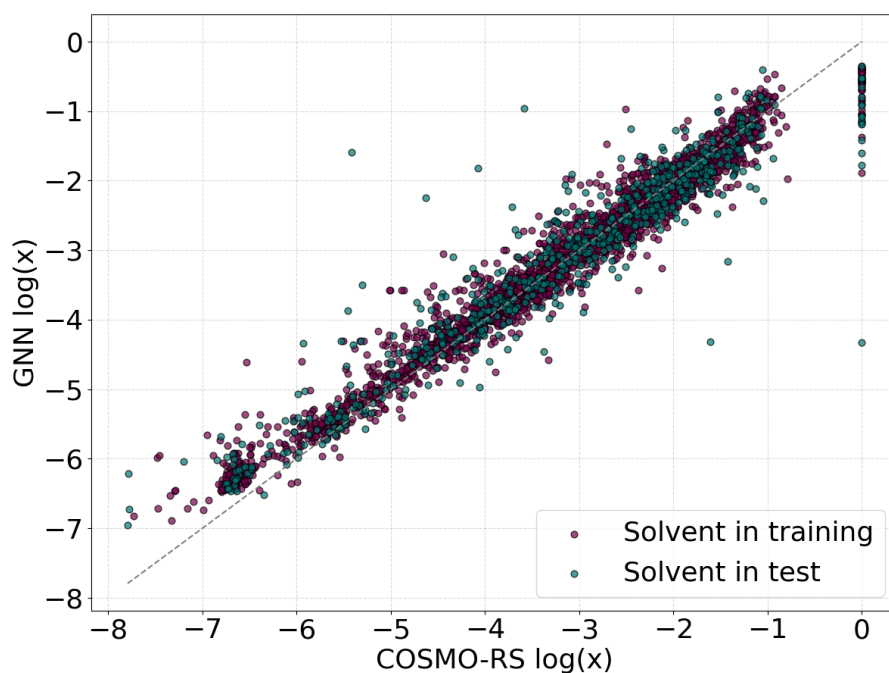


Fig. 6.4.: Parity comparison between the predicted logarithmic lignin solubility in terms of its mole fraction given by COSMO-RS and the GNN.

Fig. 6.4 shows the the parity comparison between the GNN predictions and the COSMO-RS predictions. Overall, the GNN is able to estimate the solubility values with a reasonable accuracy, achieving an R^2 of 0.9 and a MAE of 0.32 on the test set. However, a high discrepancy is observed in the case of solvents with very high solubility as predicted by COSMO-RS. It is in this precise range of high solubilities where COSMO-RS predictions tend to be less accurate [35]. This can explain the discrepancy between the GNN predictions and the COSMO-RS predicted values in this region. Despite this, the exploration of suitable solvent families for lignin can be performed by exploiting the overall correlations of the molecular structural data of all solvents.

My collaborator designed an algorithm, called PSEvolve [88], which consists in a graph-based genetic algorithm that sequentially performs graph transformations for optimizing a given criteria (in this case lignin solubility). These transformations include the addition or deletion of atoms and covalent bonds, the substitution of atoms and bonds, the relocation of atoms in the molecule and the addition of functional groups. A set of structural constraints are also considered by the PSEvolve algorithm in order to maintain the chemical feasibility after each transformation. However, for optimizing the molecular structure, an input-output mapping from the molecular structure to the lignin solubility value is needed. This is the precise role of the GNN model described in this Section.

With the task of doing solvent design for lignin solubility, the PSEvolve was initialized with a population of 1,000 molecules, and the transformations were performed over 1,000 generations producing 100 candidate solvents at each generation. Therefore, a total of 100,000 calls to the GNN were necessary to guide the optimization process of the solvent molecular structure for maximizing the lignin solubility. Since the prediction time of the GNN for each solvent is in the order of 1 millisecond (in a typical CPU machine), the entire solubility estimation process required just around a couple of minutes. By contrast, if such scheme were to be operated using COSMO-RS as the input-output mapping, and assuming that each call requires a quantum mechanical calculation that last only 30 minutes, approximately 5.7 years of computational time would be necessary. This exemplifies how GNN-based models can be also used as surrogate models of more expensive mechanistic models for accelerating the exploration of large chemical spaces.

My collaborators identified several promising solvents for lignin using this GNN-supported scheme, and verified various of these candidates experimentally [88]. Overall, the candidates selected by the PSEvolve algorithm coupled with the GNN model were indeed achieving high solubilities experimentally when compared to conventional solvents (e.g., dimethyl sulfoxide). Some discrepancies were also found in the family of ethers. For example, while the GNN predicted that both diethylene glycol dimethyl ether and diethylene glycol diethyl ether would have similar (low) lignin solubilities, only the latter prediction aligned with the experiments (8.9 wt.%) [88]. By comparison, the former species achieved a remarkable 51 wt.% solubility experimentally. This shows a clear limitation in the ability of the GNN for accurately predicting the actual behavior of distinct (but structurally similar) ethers.

Attribution on molecular graphs

In order to provide the researcher with an additional tool for the task of selecting promising solvents for lignin, the predictions of the GNN model can be used not for mere solubility estimation, but also for attributing the predictions to specific parts of the molecular graphs. In this way, the GNN model is used as an extra tool for supporting the explanation and interpretation of the results. One of such attribution techniques is the so-called integrated gradients [150]. This method satisfies two axioms, which can be applied in the realm of graphs and GNNs as follows:

- The **sensitivity axiom** establishes that given a graph \mathcal{G} and a baseline graph \mathcal{G}_{base} that differ only in one feature, and that the prediction of a GNN is different for each graph (i.e., $\text{GNN}(\mathcal{G}) \neq \text{GNN}(\mathcal{G}_{base})$), then such feature should be assigned a non-zero attribution.
- The **implementation invariance axiom** establishes that given two GNNs that are functionally equivalent (i.e., that predict the same output given the same input) their attributions should be identical, despite of the specific differences in the architecture of the GNNs.

The integrated gradients attribution for node v in graph \mathcal{G} with respect to baseline \mathcal{G}_{base} can be computed as

$$IG_v(\mathcal{G}) = (\mathbf{a}_{v,\mathcal{G}} - \mathbf{a}_{v,\mathcal{G}_{base}}) \cdot \int_1^{\alpha=0} \frac{\partial \text{GNN}(\mathcal{G}_{base} + \alpha \cdot (\mathcal{G} - \mathcal{G}_{base}))}{\partial \mathbf{a}_{v,\mathcal{G}}} \quad (6.12)$$

where, $\mathbf{a}_{v,\mathcal{G}}$ and $\mathbf{a}_{v,\mathcal{G}_{base}}$ stand for the initial node feature vector of node v in graph \mathcal{G} and graph \mathcal{G}_{base} , respectively. The operations $(\mathcal{G}_{base} + \alpha \cdot (\mathcal{G} - \mathcal{G}_{base}))$ are performed element-wise for each of the features in the graphs.

In this work, for each solvent graph, its baseline corresponds to a graph with the same connectivity, but with zeroth node features. The integrated gradients method was computed using the *Captum* library (version 0.6.0) [86]. Since, the attribution scores are intended for supporting the qualitative explanation of molecular substructures, a second GNN was trained for this purpose. This second GNN acts as a classifier between "promising" and "not-promising" solvents. The threshold used to define this two categories across the data set of 3,314 solvents was $\log(x) = -1.5$.

Fig. 6.5 shows the correspondence of the classification predictions of the second GNN and the solubility regression predictions of the first GNN. As can be seen, very few discrepancies occur across the two models, and mostly gathered around the

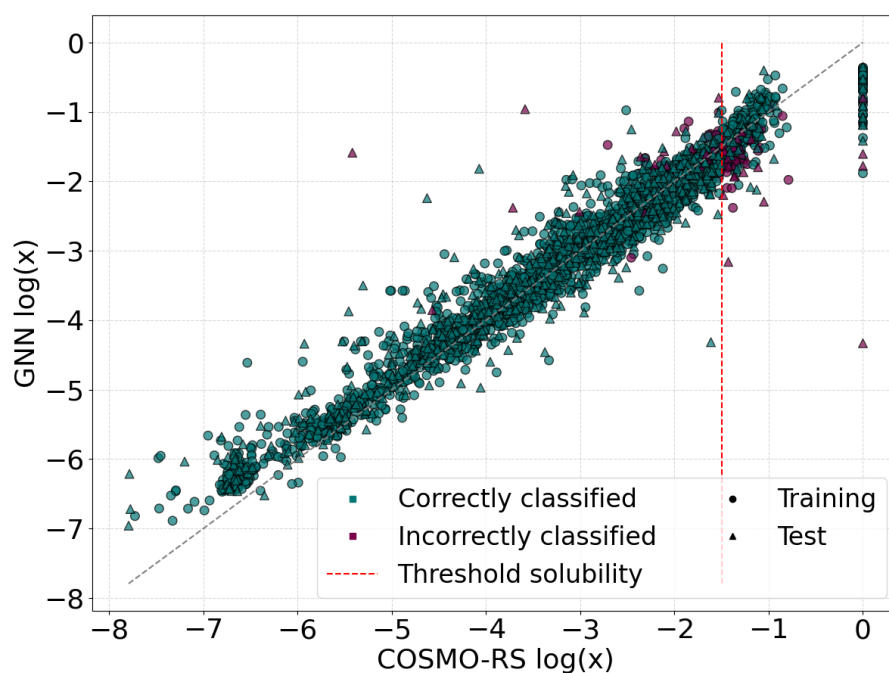


Fig. 6.5.: Correspondence of the GNN for classification and the GNN for regression for the logarithmic lignin solubility in terms of its mole fraction as predicted by COSMO-RS.

solubility threshold. Overall, 97% of the solvents were correctly classified by the GNN (94% for the test set). Moreover, around 82% of the solvents classified as “promising” are indeed classified as such by the GNN. Therefore, the second GNN (that acts as a classifier) coupled with the integrated gradients method is able to provide a relatively accurate estimation that could be used for explainability of the GNN solubility predictions.

Fig. 6.6 shows an example of the attribution scores for three molecules that were classified as “promising” by the GNN. The attribution scores are normalized for each molecule and depicted with darker color for high (importance) scores and lighter color for low (importance) score. For diethyl sulfoxide, the most important substructure for the GNN to classify it as “promising” corresponds to the sulfur atom and the attached double bond to the oxygen. Similarly, high importance is assigned to the sulfur atom in the case of thiazole. The aromatic ring, and specially the aromatic nitrogen atoms in 4-methyl-pyrimidine are selected by the GNN as the most important substructures for the prediction. In all these three cases, the predicted high solubility was confirmed experimentally by my collaborators [88].

While these scores reflect the importance that the GNN assigns to certain molecular motifs for solubility classification, it is important to highlight that they should be

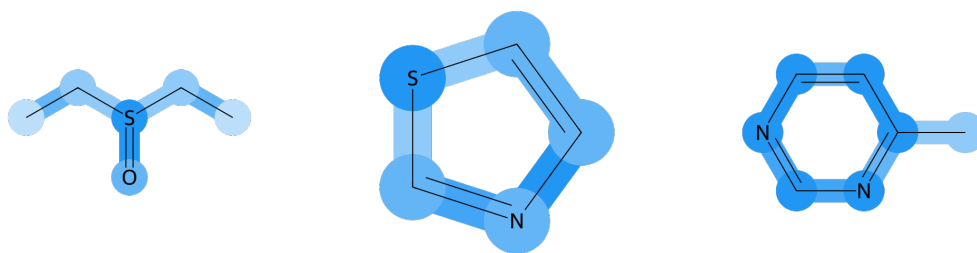


Fig. 6.6.: Depiction of the attribution scores (normalized to 0-1) for diethyl sulfoxide (left), thiazole (middle) and 4-methyl-pyrimidine (right). The magnitude of the attribution score is depicted proportional to the darkness of the color.

taken just as an extra tool to support the interpretation and exploration of the results. However, their availability could, in principle, be used to guide the chemical intuition of experts towards the selection of suitable solvents for complex systems like lignin.

6.4 Chapter summary

This Chapter provides a set of 3 case-studies where the early stages of separation process design are assisted by GNN-based models. Across these case-studies the utility of the models proposed in this dissertation is showcase at the process-level. In this way, the multi-scale framework presented in this dissertation that spans from the modeling of atoms and covalent bonds as nodes and edges in graphs is taken through the modeling of fluid phase phenomena (specifically the non-ideality of fluid phase mixtures) to finally support the tasks of the chemical engineer at the process-level scale.

The first case study covers the use of the proposed GH-GNN and GH-GNN-Margules models (introduced in Chapters 4 and 5, respectively) for the pre-selection of solvents for extractive distillation. Several representative mixtures where studied to represent challenging separations across aromatic/aliphatic, paraffin/olefin and oxygenated mixtures. The results show that the hybrid GNN models are able to suggest promising solvents in all cases, that aligned well with the available literature.

The second case study also covers the selection of solvents, but now in the context of liquid-liquid extraction for a specific system that includes the use of the ionic liquid [EMIM][BF₄] and caprolactam. In this case-study the extended GH-GNN model (introduced in Section 4.2) is used for solvent ranking and selection. Through a

collaboration led by Ann-Joelle Minor, the experimental suitability of the selected solvent (1-octanol) was confirmed. Also, in this case, the ranking of solvents obtained by the extended GH-GNN model agrees with the suggestions found in the literature for the extraction of caprolactam from aqueous mixtures.

Finally, the third case-study exemplifies the use of similar GNN-based frameworks for assisting the separation process design of complex systems, like the processing of lignin. Given that experimental data is very scarce for such type of systems, the GNN acts more as a surrogate model of more expensive mechanistic models (in this case COSMO-RS) for facilitating its use in the context of solvent design. The use of the GNN as a surrogate allowed for a significant speed-up in the solvent design process that could otherwise have taken 6 orders of magnitude more time. Similarly to the previous case-study, here the collaboration led by Laura König-Mattern allowed for the confirmation of the results experimentally, showing an overall agreement with the solvents design with the help of the GNN. Attribution is also presented as an extra tool that could support the researcher in the difficult endeavor of solvent selection.

In the future, various solvent performance metrics could be embedded into a multi-objective optimization problem in which one or more GNNs could model relevant (and yet difficult to accurately predict) properties. Moreover, the implementation of a complete framework for molecular and process design assisted by GNN-based models can be envisioned given the on-going efforts in merging common process optimization software with GNNs [170].

Conclusions

” *One never notices what has been done; one can only see what remains to be done.*

— **Marie Curie**
(Nobel Prize in Physics, 1903 and Nobel Prize
in Chemistry, 1911)

Being able to predict mixture properties from the molecular structure of its components is a problem that humanity has faced for a long time. The accurate prediction of mixture properties is not only relevant for the task of separation, but also for the task of mixture creation. The central challenge for this, resides in the enormous chemical space that models have to deal with. In principle, a model has to be predictive in nature so as to navigate the vast chemical space efficiently. This is increasingly important when considering the pressing challenges caused by the actual sustainability crisis. This also means that this type of models have to be also efficient in terms of time and resources so as to be practical for engineering applications.

One prominent example of such mixture properties is its phase equilibrium behavior. The accurate and efficient prediction of mixtures' phase equilibrium is necessary for the design and operation of novel (bio)chemical processes that are more sustainable. Such processes may span across scales, ranging from modeling the equilibrium in a living cell, to the design and operation of novel bio-refineries. This work aims at pushing the boundaries of knowledge in this precise direction. Specifically, it centers around the issue of predicting activity coefficients in an accurate and efficient manner by combining graph neural networks and mechanistic understanding into a single hybrid approach.

7.1 Summary

Starting from the simplest case of predicting isothermal activity coefficients at infinite dilution, Chapter 3 shows that GNN-based models are able to effectively predict

isothermal IDACs across a relatively large chemical space. The accuracy of such GNN-based models outperform that of models that have been the popular choice for decades (e.g. UNIFAC and COSMO-RS). More interestingly, Chapter 3 also shows, via a hybrid parallel approach, that these traditional methods incur in systematic mismatches that can be effectively learn from the component's molecular structure alone. As a result, the proposed hybrid parallel GNN models show a better predictive accuracy compared to the most popular phenomenological models.

Chapter 4 expands the use of this GNN framework towards the modeling of IDACs at varying temperatures. This is achieved by the introduction of an expression derived from the Gibbs-Helmholtz relation that controls the temperature dependency of activity coefficients. The extensive testing of the resulting model, called Gibbs-Helmholtz Graph Neural Network (GH-GNN) shows a more accurate predictions across various temperature ranges and chemical classes than the popular UNIFAC-Dortmund model, specially within the space of mixtures delimited by the available training data. In this endeavor, extensive data curation and digitization was carried out to promote the openness of research and the advancement of the field that could only come from researchers and engineers around the globe despite their resource availability.

Two extensions of the GH-GNN model are also studied in Chapter 4. The first, towards including systems with ionic liquids. This extended GH-GNN model shows better predictive performance than one of the latest versions of UNIFAC for ionic liquids. And the second, towards mixtures of polymer solutions. Here, the extended GH-GNN model also shows promising performance compared to UNIFAC-based models specifically tailored for predicting this type of mixtures. Both of these cases (i.e., mixtures with ionic liquids and polymers) are relevant for the discovery and development of better separation processes.

Following this logical path, Chapter 5 takes the last step in generalizing the model to activity coefficients at varying temperatures and compositions. This is achieved by coupling the extended Margules equation with the proposed GH-GNN. The resulting serial hybrid model is able to reproduce the vapor-liquid equilibria behavior of many types of systems. Extensive binary mixtures were studied and some ternary mixtures were also included in the analysis. While in this case UNIFAC-Dortmund overall outperforms the performance of the GH-GNN-Margules model, the comparison has to be made in the light of the type and amount of data that was accessible for model development. On one hand, the UNIFAC-Dortmund model was parameterized with extensive experimental data spanning vapor-liquid, liquid-liquid equilibria, azeotropic, caloric and infinite dilution data. By contrast, the proposed GH-GNN-

Margules framework only utilizes information at infinite dilution. This highlights a critical point regarding the data accessibility: if better predictive models have to be developed, the current issue of difficult data accessibility needs to be resolved.

The dissertation concludes with a series of case-studies on Chapter 6, where the utility of the proposed GNN-based models is exemplified for the relevant scenarios of solvent pre-selection for extractive distillation and liquid-liquid extraction. Moreover, the use of such type of GNN-based models is exemplified for the case of assisting the task of solvent design for a lignin process. In all these scenarios the GNN-based models allowed not only for an accurate estimation of the required mixture properties, but perhaps more importantly, for the exploration of extensive chemical spaces that could not be (as effectively) explored with traditional approaches.

7.2 Outlook

This dissertation extensively explored the use of GNNs for the prediction of activity coefficients. Not only the use of standalone GNNs was studied, but also the hybrid combination of GNNs with phenomenological or mechanistic models was analyzed into two main arrangements: parallel and serial. This area of research is expected to be highly relevant in the near future. The constant advancements in the realm of machine learning and hardware development provides the chemical engineer with an unprecedented myriad of tools that could assist and accelerate the solution of pressing challenges. This requires the expertise of chemical engineers that are able to fusion the power of chemical engineering mechanistic modeling and understanding with the flexibility and efficiency of modern data-driven methods.

It is my believe that, just like Prof. Roger Sargent foresaw the potential of computational methods for aiding process engineering, even when computing power was far more restricted than it is today, modern machine learning techniques will inevitably make a huge impact on process systems engineering overall. Specifically, for advancing the accurate and efficient prediction of mixture properties. In this path, there are several challenges. The first one being the educational challenge that modern and future chemical engineers have to bear. They not only need to master the fundamentals (e.g., thermodynamics, reaction, mass and energy transfer) but also they need to cope with the ability to quickly adapt modern tools from computer science to solve relevant problems. The second challenge is the availability and decentralization of physicochemical experimental data. If data persists to be in the hands of the minority, the advancement will not be as fast and abundant as it could

otherwise be. The culture of the chemical engineering community needs to change towards a more open and collaborative one if the benefits of the collective “mind” are desired. Third, enforcing physical constraints into data-driven models is not yet straightforward. The development of new hybrid modeling approaches is envisioned in the form of physical prior injections and mechanistic knowledge embedding.

Overall, this is an exciting time for the modern chemical engineer. The unmovable physical constraints of our Universe need to be mixed with the highly flexible (yet powerful) data-driven models. Perhaps, these type of “mixture” models are the ones needed to finally conquer the accurate and efficient prediction of how real mixtures behave. Mixtures, everywhere present...

Bibliography

- [1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, et al. “Tensorflow: Large-scale machine learning on heterogeneous distributed systems”. In: *arXiv:1603.04467* (2016) (cit. on p. 4).
- [2] Michael H. Abraham. “Scales of solute hydrogen-bonding: their construction and application to physicochemical and biochemical processes”. In: *Chemical Society Reviews* 22.2 (1993), pp. 73–83 (cit. on pp. 24, 181).
- [3] Denis S. Abrams and John M. Prausnitz. “Statistical thermodynamics of liquid mixtures: a new expression for the excess Gibbs energy of partly or completely miscible systems”. In: *AIChE Journal* 21.1 (1975), pp. 116–128 (cit. on pp. 3, 16, 18).
- [4] Subhash Ajmani, Stephen C. Rogers, Mark H. Barley, Andrew N. Burgess, and David J. Livingstone. “Characterization of mixtures part 1: Prediction of infinite-dilution activity coefficients using neural network-based QSPR models”. In: *QSAR & Combinatorial Science* 27.11-12 (2008), pp. 1346–1361 (cit. on p. 25).
- [5] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. “Optuna: A next-generation hyperparameter optimization framework”. In: *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2019, pp. 2623–2631 (cit. on pp. 53, 72, 190, 192).
- [6] David J. Andersen and Donald H. Lindsley. “A valid Margules formulation for an asymmetric ternary solution: revision of the olivine-ilmenite thermometer, with applications”. In: *Geochimica et Cosmochimica Acta* 45.6 (1981), pp. 847–853 (cit. on p. 196).
- [7] Evan R. Antoniuk, Peggy Li, Bhavya Kailkhura, and Anna M. Hiszpanski. “Representing polymers as periodic graphs with learned descriptors for accurate polymer property predictions”. In: *Journal of Chemical Information and Modeling* 62.22 (2022), pp. 5435–5445 (cit. on pp. 88, 89).
- [8] Adem R. N. Aouichaoui, Fan Fan, Jens Abildskov, and Gürkan Sin. “Application of interpretable group-embedded graph neural networks for pure compound properties”. In: *Computers & Chemical Engineering* 176 (2023), p. 108291 (cit. on p. 3).
- [9] Zadjia Atik, Detlef Gruber, Michael Krummen, and Jürgen Gmehling. “Measurement of activity coefficients at infinite dilution of benzene, toluene, ethanol, esters, ketones, and ethers at various temperatures in water using the dilutor technique”. In: *Journal of Chemical & Engineering Data* 49.5 (2004), pp. 1429–1432 (cit. on p. 61).

- [10] Indra Bahadur, Byron Bradley Govender, Khalid Osman, Mark D. Williams-Wynn, Wayne Michael Nelson, Paramespri Naidoo, and Deresh Ramjugernath. "Measurement of activity coefficients at infinite dilution of organic solutes in the ionic liquid 1-ethyl-3-methylimidazolium 2-(2-methoxyethoxy) ethylsulfate at T=(308.15, 313.15, 323.15 and 333.15) K using gas + liquid chromatography". In: *The Journal of Chemical Thermodynamics* 70 (2014), pp. 245–252 (cit. on p. 49).
- [11] Joao C. Bastos, Manuela E. Soares, and Augusto G. Medina. "Selection of solvents for extractive distillation. A data bank for activity coefficients at infinite dilution". In: *Industrial & Engineering Chemistry Process Design and Development* 24.2 (1985), pp. 420–426 (cit. on p. 17).
- [12] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, et al. "Relational inductive biases, deep learning, and graph networks". In: *arXiv:1806.01261* (2018) (cit. on p. 62).
- [13] Hesam Ahmadian Behrooz and R. Bozorgmehry Boozarjomehry. "Prediction of limiting activity coefficients for binary vapor-liquid equilibrium using neural networks". In: *Fluid Phase Equilibria* 433 (2017), pp. 174–183 (cit. on p. 25).
- [14] Caleb Bell, Yoel Rene Cortes-Pena, and Contributors. *Chemicals: Chemical properties component of chemical engineering Design Library (ChEDL)*. <https://github.com/CalebBell/chemicals>. Accessed: March 2024 (cit. on p. 117).
- [15] Lloyd Berg. *Separation of benzene from close boiling hydrocarbons by extractive distillation*. US Patent 5,458,741. 1995 (cit. on pp. 124, 125).
- [16] Lloyd Berg. *Separation of benzene from non-aromatic hydrocarbons by extractive distillation*. US Patent 4,514,262. 1985 (cit. on p. 123).
- [17] Lloyd Berg. *Separation of heptane from 1-heptene by extractive distillation*. US Patent 5,443,697. 1995 (cit. on p. 125).
- [18] Alain Berthod, Maria Jose Ruiz-Angel, and Samuel Carda-Broch. "Recent advances on ionic liquid uses in separation techniques". In: *Journal of Chromatography A* 1559 (2018), pp. 2–16 (cit. on p. 82).
- [19] Christopher M. Bishop. *Pattern recognition and machine learning*. Springer New York, 2006, p. 778 (cit. on p. 21).
- [20] Marek Blahušiak, Anton A. Kiss, Katarina Babic, Sascha R. A. Kersten, Gerrald Barge-man, and Boelo Schuur. "Insights into the selection and design of fluid separation processes". In: *Separation and Purification Technology* 194 (2018), pp. 301–318 (cit. on p. 119).
- [21] Beatriz Blanco, Maria Teresa Sanz, Sagrario Beltrán, José Luis Cabezas, and José Coca. "Vapor-liquid equilibria for the ternary system benzene + n-heptane + n,n-dimethylformamide at 101.33 kPa". In: *Fluid Phase Equilibria* 175.1-2 (2000), pp. 117–124 (cit. on p. 110).
- [22] Arnold Aaron Bondi. *Physical properties of molecular crystals, liquids, and glasses*. John Wiley & Sons, Inc., 1968, p. 502 (cit. on pp. vii, 3).

- [23] Esteban Alberto Brignole and Selva Pereda. *Phase equilibrium engineering*. Elsevier, 2013, p. 346 (cit. on p. 5).
- [24] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. “Geometric deep learning: Grids, groups, graphs, geodesics, and gauges”. In: *arXiv:2104.13478* (2021) (cit. on p. 20).
- [25] Thomas Brouwer, Sascha R. A. Kersten, Gerrald Bargeman, and Boelo Schuur. “Solvent pre-selection for extractive distillation using infinite dilution activity coefficients and the three-component Margules equation”. In: *Separation and Purification Technology* 276 (2021), p. 119230 (cit. on pp. 17, 118, 119).
- [26] Thomas Brouwer, Sascha R. A. Kersten, Gerrald Bargeman, and Boelo Schuur. “Trends in solvent impact on infinite dilution activity coefficients of solutes reviewed and visualized using an algorithm to support selection of solvents for greener fluid separations”. In: *Separation and Purification Technology* 272 (2021), p. 118727 (cit. on pp. 49, 61, 79).
- [27] Thomas Brouwer and Boelo Schuur. “Erratum for “Model performances evaluated for infinite dilution activity coefficients prediction at 298.15 K””. In: *Industrial & Engineering Chemistry Research* 62.14 (2023), pp. 6016–6017 (cit. on p. 25).
- [28] Thomas Brouwer and Boelo Schuur. “Model performances evaluated for infinite dilution activity coefficients prediction at 298.15 K”. In: *Industrial & Engineering Chemistry Research* 58.20 (2019), pp. 8903–8914 (cit. on pp. 23, 25, 34).
- [29] Roger M. Butler and John A. Bichard. *Separation of aromatics from hydrocarbon streams*. US Patent 3,114,783. 1963 (cit. on p. 123).
- [30] Tianle Cai, Shengjie Luo, Keyulu Xu, Di He, Tie-yan Liu, and Liwei Wang. “Graph-norm: A principled approach to accelerating graph neural network training”. In: *International Conference on Machine Learning*. PMLR, 2021, pp. 1204–1215 (cit. on pp. 67, 68).
- [31] Cecil O. Carter. *Separating olefins from paraffins with dimethyl sulfoxide extractant*. US Patent 4,267,034. 1981 (cit. on p. 125).
- [32] Guzhang Chen, Zhen Song, Zhiwen Qi, and Kai Sundmacher. “Neural recommender system for the activity coefficient prediction and UNIFAC model extension of ionic liquid-solute systems”. In: *AIChE Journal* 67.4 (2021), e17171 (cit. on pp. 25, 82–85, 93, 129, 131).
- [33] Julie Damay, Fabian Jirasek, Marius Kloft, Michael Bortz, and Hans Hasse. “Predicting activity coefficients at infinite dilution for varying temperatures by matrix completion”. In: *Industrial & Engineering Chemistry Research* 60.40 (2021), pp. 14564–14578 (cit. on pp. 25, 49, 50, 61, 70, 72, 73, 77, 81, 91).
- [34] Julie Damay, Gleb Ryzhakov, Fabian Jirasek, Hans Hasse, Ivan Oseledets, and Michael Bortz. “Predicting temperature-dependent activity coefficients at infinite dilution using tensor completion”. In: *Chemie Ingenieur Technik* (2023) (cit. on pp. 25, 91).
- [35] Dassault Systèmes company. *COSMOtherm Reference Manual*. 2019 (cit. on p. 134).

- [36] George Edward Davis. *A handbook of chemical engineering illustrated with working examples and numerous drawings from actual installations*. Davis Bros, 1904 (cit. on p. 2).
- [37] Jean-Charles De Hemptinne, Georgios M. Kontogeorgis, Ralf Dohrn, Ioannis G. Economou, Antoon Ten Kate, Susanna Kuitunen, Ljudmila Fele Žilnik, Maria Grazia De Angelis, et al. “A view on the future of applied thermodynamics”. In: *Industrial & Engineering Chemistry Research* 61.39 (2022), pp. 14664–14680 (cit. on pp. 17, 85).
- [38] Yannick Djoumbou Feunang, Roman Eisner, Craig Knox, Leonid Chepelev, Janna Hastings, Gareth Owen, Eoin Fahy, Christoph Steinbeck, et al. “ClassyFire: automated chemical classification with a comprehensive, computable taxonomy”. In: *Journal of Cheminformatics* 8.1 (2016), pp. 1–20 (cit. on pp. 50, 61, 186, 188, 189).
- [39] V. Dohnal. “14 Measurement of limiting activity coefficients using analytical tools”. In: *Experimental Thermodynamics*. Vol. 7. Elsevier, 2005, pp. 359–381 (cit. on p. 49).
- [40] Urszula Domańska, Monika Karpińska, Anna Wiśniewska, and Zbigniew Dabrowski. “Ammonium ionic liquids in extraction of bio-butan-1-ol from water phase using activity coefficients at infinite dilution”. In: *Fluid Phase Equilibria* 479 (2019), pp. 9–16 (cit. on p. 49).
- [41] Dortmund Data Bank. www.ddbst.com. Accessed: December 2021 (cit. on p. 48).
- [42] Frank Eckert and Andreas Klamt. “Fast solvent screening via quantum chemistry: COSMO-RS approach”. In: *AIChE Journal* 48.2 (2002), pp. 369–385 (cit. on p. 24).
- [43] Ernesto Estrada, Gerardo A. Díaz, and Eduardo J. Delgado. “Predicting infinite dilution activity coefficients of organic compounds in water by quantum-connectivity descriptors”. In: *Journal of Computer-Aided Molecular Design* 20.9 (2006), pp. 539–548 (cit. on p. 25).
- [44] Andrew L. Fanning, Daniel W. O’Neill, Jason Hickel, and Nicolas Roux. “The social shortfall and ecological overshoot of nations”. In: *Nature Sustainability* 5.1 (2022), pp. 26–36 (cit. on p. 1).
- [45] Kobi C. Felton, Hashem Ben-Safar, and 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 (cit. on p. 25).
- [46] Matthias Fey and Jan Eric Lenssen. “Fast graph representation learning with PyTorch Geometric”. In: *arXiv:1903.02428* (2019) (cit. on pp. 27, 34).
- [47] Aage Fredenslund, Russell L. Jones, and John M. Prausnitz. “Group-contribution estimation of activity coefficients in nonideal liquid mixtures”. In: *AIChE Journal* 21.6 (1975), pp. 1086–1099 (cit. on pp. 24, 44, 178).
- [48] Joseph C. Gentry, Lloyd Berg, John C. McIntyre, and Randa W. Wytcherley. *Process to recover benzene from mixed hydrocarbons by extractive distillation*. US Patent 5,399,244. 1995 (cit. on pp. 123, 124).

- [49] Christoph Gertig, Kai Leonhard, and André Bardow. “Computer-aided molecular and processes design based on quantum chemistry: current status and future prospects”. In: *Current Opinion in Chemical Engineering* 27 (2020), pp. 89–97 (cit. on p. 2).
- [50] Roland Geyer, Jenna R. Jambeck, and Kara Lavender Law. “Production, use, and fate of all plastics ever made”. In: *Science Advances* 3.7 (2017), e1700782 (cit. on p. 85).
- [51] J. Willard Gibbs. “On the equilibrium of heterogeneous substances”. In: *Transactions of the Connecticut Academy of Arts and Sciences* 3 (1878). Published in parts, pp. 108–248, 343–524 (cit. on p. 13).
- [52] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. “Neural message passing for quantum chemistry”. In: *International Conference on Machine Learning*. PMLR, 2017, pp. 1263–1272 (cit. on pp. 21, 27, 30, 69).
- [53] Francesc Giralt, G. Espinosa, A. Arenas, J. Ferre-Gine, Ll. Amat, X. Girones, R. Carbó-Dorca, and Y. Cohen. “Estimation of infinite dilution activity coefficients of organic compounds in water with neural classifiers”. In: *AIChE Journal* 50.6 (2004), pp. 1315–1343 (cit. on p. 25).
- [54] Jarka Glassey and Moritz Von Stosch. *Hybrid modeling in process industries*. CRC Press, 2018, p. 231 (cit. on pp. 18, 19, 41).
- [55] J. Gmehling, D. Tiegs, A. Medina, M. Soares, J. Bastos, P. Alessi, I. Kikic, M. Schiller, et al. *Activity coefficients at infinite dilution*. Vol. IX. DECHEMA Chemistry Data Series, 2008 (cit. on pp. 48, 49, 184).
- [56] Jürgen Gmehling, Michael Kleiber, Bärbel Kolbe, and Jürgen Rarey. *Chemical thermodynamics for process simulation*. John Wiley & Sons, 2019 (cit. on pp. 4, 5, 10, 11, 34, 96–98).
- [57] Jürgen Gmehling, Jiding Li, and Martin Schiller. “A modified UNIFAC model. 2. Present parameter matrix and results for different thermodynamic properties”. In: *Industrial & Engineering Chemistry Research* 32.1 (1993), pp. 178–193 (cit. on pp. 24, 178).
- [58] Jürgen Gmehling, Jürgen Lohmann, Antje Jakob, Jiding Li, and Ralph Joh. “A modified UNIFAC (Dortmund) model. 3. Revision and extension”. In: *Industrial & Engineering Chemistry Research* 37.12 (1998), pp. 4876–4882 (cit. on pp. 24, 178).
- [59] Jürgen Gmehling, Roland Wittig, Jürgen Lohmann, and Ralph Joh. “A modified UNIFAC (Dortmund) model. 4. Revision and extension”. In: *Industrial & Engineering Chemistry Research* 41.6 (2002), pp. 1678–1688 (cit. on pp. 24, 45, 93, 103, 104, 114, 178).
- [60] O. Großmann, D. Bellaire, Nicolas Hayer, Fabian Jirasek, and Hans Hasse. “Prediction of diffusion coefficients at infinite dilution by matrix completion”. In: *Chemie Ingenieur Technik* 94.9 (2022), pp. 1354–1354 (cit. on p. 25).

- [61] Oliver Großmann, Daniel Bellaire, Nicolas Hayer, Fabian Jirasek, and Hans Hasse. “Database for liquid phase diffusion coefficients at infinite dilution at 298 K and matrix completion methods for their prediction”. In: *Digital Discovery* 1.6 (2022), pp. 886–897 (cit. on p. 25).
- [62] Mohamed K. Hadj-Kali, Abdullah M. Al-Anazi, Saeed M. Alhawtali, and Irfan Wazeer. “Liquid-liquid separation of n-hexane/1-hexene and cyclohexane/cyclohexene using deep eutectic solvents”. In: *Journal of Molecular Liquids* 344 (2021), p. 117776 (cit. on p. 125).
- [63] Jan Haidl and Vladimír Dohnal. “Activity coefficients of water at infinite dilution in common oxygenated solvents”. In: *Journal of Chemical & Engineering Data* 65.5 (2020), pp. 2790–2797 (cit. on p. 61).
- [64] Charles M. Hansen. *Hansen solubility parameters: A user’s handbook*. CRC press, 2007, p. 544 (cit. on p. 117).
- [65] Charles M. Hansen. *The three dimensional solubility parameter*. Vol. 14. Copenhagen Danish Technical Press, 1967 (cit. on pp. 24, 176).
- [66] Wen Hao, H. S. Elbro, and P. Alessi. *DECHEMA Chemistry Data Series Vol. XIV. Polymer Solution Data Collection*. 1993 (cit. on pp. 87, 193).
- [67] Nicolas Hayer, Fabian Jirasek, and Hans Hasse. “Prediction of Henry’s law constants by matrix completion”. In: *AIChE Journal* 68.9 (2022), e17753 (cit. on p. 25).
- [68] W.M. Haynes. *CRC handbook of chemistry and physics*. Vol. 94. CRC press, 2013 (cit. on p. 63).
- [69] Tim Head, Manoj Kumar, Holger Nahrstaedt, Gilles Louppe, and Iaroslav Shcherbatyi. *scikit-optimize/scikit-optimize*. Version v0.9.0. Accessed: November 2021 (cit. on p. 182).
- [70] Toshihiko Hiak, Kiyofumi Kurihara, and Kazuo Kojima. “Vapor-liquid equilibria for acetone + chloroform + methanol and constituent binary systems at 101.3 kPa”. In: *Journal of Chemical and Engineering Data* 39.4 (1994), pp. 714–719 (cit. on p. 110).
- [71] Joel H. Hildebrand. “Thermodynamic aspects of the theory of non-electrolytic solutions.” In: *Chemical Reviews* 18.2 (1936), pp. 315–323 (cit. on pp. 24, 176).
- [72] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. “Improving neural networks by preventing co-adaptation of feature detectors”. In: *arXiv:1207.0580* (2012) (cit. on p. 31).
- [73] Sergey Ioffe and Christian Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: *International Conference on Machine Learning*. PMLR. 2015, pp. 448–456 (cit. on p. 31).
- [74] Fabian Jirasek, Rodrigo A. S. Alves, Julie Damay, Robert A. Vandermeulen, Robert Bamler, Michael Bortz, Stephan Mandt, Marius Kloft, et al. “Machine learning in thermodynamics: Prediction of activity coefficients by matrix completion”. In: *The Journal of Physical Chemistry Letters* 11.3 (2020), pp. 981–985 (cit. on pp. 25, 91).

- [75] Fabian Jirasek, Robert Bamler, and Stephan Mandt. “Hybridizing physical and data-driven prediction methods for physicochemical properties”. In: *Chemical Communications* 56.82 (2020), pp. 12407–12410 (cit. on pp. 25, 44, 45, 55).
- [76] Fabian Jirasek and Hans Hasse. “Combining machine learning with physical knowledge in thermodynamic modeling of fluid mixtures”. In: *Annual Review of Chemical and Biomolecular Engineering* 14 (2023), pp. 31–51 (cit. on pp. 9, 17).
- [77] Fabian Jirasek and Hans Hasse. “Perspective: machine learning of thermophysical properties”. In: *Fluid Phase Equilibria* 549 (2021), p. 113206 (cit. on p. 9).
- [78] Kevin G. Joback and Robert C. Reid. “Estimation of pure-component properties from group-contributions”. In: *Chemical Engineering Communications* 57.1-6 (1987), pp. 233–243 (cit. on p. 117).
- [79] Akio Kamimura, Yuto Shiramatsu, and Takuji Kawamoto. “Depolymerization of polyamide 6 in hydrophilic ionic liquids”. In: *Green Energy & Environment* 4.2 (2019), pp. 166–170 (cit. on pp. 127, 129).
- [80] Andrei Kazakov, Joe Magee, Robert Chirico, Vladimir Diky, Kenneth Kroenlein, Chris Muzny, and Michael Frenkel. *Ionic liquids database - ILThermo (v2.0)*. Accessed: February 2024 (cit. on p. 82).
- [81] I. Kikic, P. Alessi, Peter Rasmussen, and Aage Fredenslund. “On the combinatorial part of the UNIFAC and UNIQUAC models”. In: *The Canadian Journal of Chemical Engineering* 58.2 (1980), pp. 253–258 (cit. on pp. 24, 178).
- [82] Thomas N. Kipf and Max Welling. “Semi-supervised classification with graph convolutional networks”. In: *arXiv:1609.02907* (2016) (cit. on p. 21).
- [83] Andreas Klamt, Frank Eckert, and Wolfgang Arlt. “COSMO-RS: an alternative to simulation for calculating thermodynamic properties of liquid mixtures”. In: *Annual review of chemical and biomolecular engineering* 1 (2010), pp. 101–122 (cit. on p. 3).
- [84] Kazuo Kojima, Katsumi Tochigi, Kiyofumi Kurihara, and Mikiyoshi Nakamichi. “Isobaric vapor-liquid equilibria for acetone + chloroform + benzene and the three constituent binary systems”. In: *Journal of Chemical and Engineering Data* 36.3 (1991), pp. 343–345 (cit. on p. 110).
- [85] Kazuo Kojima, Suojiang Zhang, and Toshihiko Hiaki. “Measuring methods of infinite dilution activity coefficients and a database for systems including water”. In: *Fluid Phase Equilibria* 131.1-2 (1997), pp. 145–179 (cit. on p. 79).
- [86] Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, et al. “Captum: A unified and generic model interpretability library for PyTorch”. In: *arXiv:2009.07896* (2020) (cit. on p. 136).
- [87] Laura König-Mattern, Anastasia O. Komarova, Arpa Ghosh, Steffen Linke, Liisa Rihko-Struckmann, Jeremy Luterbacher, and Kai Sundmacher. “High-throughput computational solvent screening for lignocellulosic biomass processing”. In: *Chemical Engineering Journal* 452 (2023), p. 139476 (cit. on p. 132).

- [88] Laura König-Mattern, Edgar Ivan Sanchez Medina, Anastasia O. Komarova, Steffen Linke, Liisa Rihko-Struckmann, Jeremy S. Luterbacher, and Kai Sundmacher. “Machine learning-supported solvent design for lignin-first biorefineries and lignin upgrading”. In: *Chemical Engineering Journal* 495 (2024), p. 153524 (cit. on pp. 132, 135, 137).
- [89] Georgios M. Kontogeorgis, Ralf Dohrn, Ioannis G. Economou, Jean-Charles de Hemptinne, Antoon Ten Kate, Susanna Kuitunen, Miranda Mooijer, Ljudmila Fele Zilnik, et al. “Industrial requirements for thermodynamic and transport properties: 2020”. In: *Industrial & Engineering Chemistry Research* 60.13 (2021), pp. 4987–5013 (cit. on p. 85).
- [90] Georgios M. Kontogeorgis, Aage Fredenslund, and Dimitrios P. Tassios. “Simple activity coefficient model for the prediction of solvent activities in polymer solutions”. In: *Industrial & Engineering Chemistry Research* 32.2 (1993), pp. 362–372 (cit. on pp. 87, 88, 91–93).
- [91] *Korean Data Bank*. <https://www.cheric.org/research/kdb>. Accessed: 01 March 2024 (cit. on pp. 98, 99).
- [92] Kiyofumi Kurihara, Hiroaki Hori, and Kazuo Kojima. “Vapor-liquid equilibrium data for acetone + methanol + benzene, chloroform + methanol + benzene, and constituent binary systems at 101.3 kPa”. In: *Journal of Chemical & Engineering Data* 43.2 (1998), pp. 264–268 (cit. on pp. 110, 111).
- [93] Oak Ridge National Laboratory. *Materials for separation technologies. Energy and emission reduction opportunities*. <https://www.osti.gov/biblio/1218755>. 2005 (cit. on p. 1).
- [94] Irving Langmuir. “The distribution and orientation of molecules”. In: *Third Colloid Symposium Monograph* 3 (1925), pp. 48–75 (cit. on p. 3).
- [95] Bent L. Larsen, Peter Rasmussen, and Aage Fredenslund. “A modified UNIFAC group-contribution model for prediction of phase equilibria and heats of mixing”. In: *Industrial & Engineering Chemistry Research* 26.11 (1987), pp. 2274–2286 (cit. on pp. 24, 44, 178).
- [96] Michael J. Lazzaroni, David Bush, Charles A. Eckert, Timothy C. Frank, Sumnesh Gupta, and James D. Olson. “Revision of MOSCED parameters and extension to solid solubility calculations”. In: *Industrial & Engineering Chemistry Research* 44.11 (2005), pp. 4075–4083 (cit. on p. 60).
- [97] Zhigang Lei, Biaohua Chen, and Zhongwei Ding. “Chapter 2 - Extractive distillation”. In: *Special Distillation Processes*. Ed. by Zhigang Lei, Biaohua Chen, and Zhongwei Ding. Amsterdam: Elsevier Science, 2005, pp. 59–144 (cit. on p. 117).
- [98] Zhigang Lei, Chengyue Li, and Biaohua Chen. “Extractive distillation: A review”. In: *Separation & Purification Reviews* 32.2 (2003), pp. 121–213 (cit. on p. 116).
- [99] Zhigang Lei, Rongqi Zhou, and Zhanting Duan. “Process improvement on separating C₄ by extractive distillation”. In: *Chemical Engineering Journal* 85.2-3 (2002), pp. 379–386 (cit. on p. 125).

- [100]Jean-Claude Lerol, Jean-Claude Masson, Henri Renon, Jean-Francois Fabries, and Henri Sannier. “Accurate measurement of activity coefficient at infinite dilution by inert gas stripping and gas chromatography”. In: *Industrial & Engineering Chemistry Process Design and Development* 16.1 (1977), pp. 139–144 (cit. on p. 49).
- [101]Ilya Loshchilov and Frank Hutter. “Decoupled weight decay regularization”. In: *arXiv:1711.05101* (2017) (cit. on p. 72).
- [102]Łukasz Marcinkowski, Joachim Eichenlaub, Elham Ghasemi, Żaneta Polkowska, and Adam Kloskowski. “Measurements of activity coefficients at infinite dilution for organic solutes in the ionic liquids n-ethyl- and n-octyl-n-methylmorpholinium bis(trifluoromethanesulfonyl) imide. A useful tool for solvent selection”. In: *Molecules* 25.3 (2020), p. 634 (cit. on p. 49).
- [103]Kevin McBride, Edgar Ivan Sanchez Medina, and Kai Sundmacher. “Hybrid semi-parametric modeling in separation processes: A review”. In: *Chemie Ingenieur Technik* 92.7 (2020), pp. 842–855 (cit. on pp. 4, 9, 17–19, 41, 97).
- [104]Warren L. McCabe and E. W. Thiele. “Graphical design of fractionating columns”. In: *Industrial & Engineering Chemistry* 17.6 (1925), pp. 605–611 (cit. on p. 2).
- [105]Ann-Joelle Minor, Edgar Ivan Sanchez Medina, Ruben Goldhahn, Steffen Linke, Liisa Rihko-Struckmann, and Kai Sundmacher. *Can ionic liquids compete with other solvents in the chemical recycling of Nylon 6?* Under review. 2024 (cit. on pp. 129–131).
- [106]Brooke E. Mitchell and Peter C. Jurs. “Prediction of infinite dilution activity coefficients of organic compounds in aqueous solution from molecular structure”. In: *Journal of Chemical Information and Computer Sciences* 38.2 (1998), pp. 200–209 (cit. on p. 25).
- [107]Hirotomo Moriwaki, Yu-Shi Tian, Norihito Kawashita, and Tatsuya Takagi. “Mordred: a molecular descriptor calculator”. In: *Journal of Cheminformatics* 10.1 (2018), pp. 1–14 (cit. on p. 63).
- [108]Biswajit Mukhopadhyay, Sabyasachi Basu, and Michael J. Holdaway. “A discussion of Margules-type formulations for multicomponent solutions with a generalized approach”. In: *Geochimica et Cosmochimica Acta* 57.2 (1993), pp. 277–283 (cit. on p. 96).
- [109]Russell L. Nielsen and James H. Weber. “Vapor-liquid equilibria at subatmospheric pressures. Binary and ternary systems containing ethyl alcohol, benzene, and n-heptane.” In: *Journal of Chemical and Engineering Data* 4.2 (1959), pp. 145–151 (cit. on p. 110).
- [110]Takeru Oishi and John M. Prausnitz. “Estimation of solvent activities in polymer solutions using a group-contribution method”. In: *Industrial & Engineering Chemistry Process Design and Development* 17.3 (1978), pp. 333–339 (cit. on pp. 85, 88).
- [111]*OPSiN (Open Parser for Systematic IUPAC Nomenclature)*. <https://opsin.ch.cam.ac.uk/>. Accessed: 01 March 2024 (cit. on p. 99).

- [112]P. Oracz and S. Warycha. "Vapour-liquid equilibria. VII: The ternary system hexane + methanol + acetone at 313.15 K". In: *Fluid Phase Equilibria* 108.1-2 (1995), pp. 199–211 (cit. on pp. 110, 112).
- [113]Kamil Padaszynski. "In silico calculation of infinite dilution activity coefficients of molecular solutes in ionic liquids: critical review of current methods and new models based on three machine learning algorithms". In: *Journal of Chemical Information and Modeling* 56.8 (2016), pp. 1420–1437 (cit. on p. 25).
- [114]Costas Panayiotou, Ioannis Zuburtikudis, and Hadil Abu Khalifeh. "Linear Free-Energy Relationships and solvation thermodynamics: The Thermodynamic basis of LFER linearity". In: *Industrial & Engineering Chemistry Research* 62.6 (2023), pp. 2989–3000 (cit. on p. 181).
- [115]Georgia D. Pappa, Epaminondas C. Voutsas, and Dimitrios P Tassios. "Prediction of activity coefficients in polymer and copolymer solutions using simple activity coefficient models". In: *Industrial & Engineering Chemistry Research* 38.12 (1999), pp. 4975–4984 (cit. on pp. 91, 92).
- [116]Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, et al. "PyTorch: An imperative style, high-performance deep learning library". In: *Advances in Neural Information Processing Systems* 32 (2019) (cit. on pp. 4, 34).
- [117]D. Patterson, Y. B. Tewari, H. P. Schreiber, and J. E. Guillet. "Application of gas-liquid chromatography to the thermodynamics of polymer solutions". In: *Macromolecules* 4.3 (1971), pp. 356–359 (cit. on pp. 85, 86).
- [118]Nicholas A. Peppas. *One hundred years of chemical engineering: from Lewis M. Norton (MIT 1888) to present*. Vol. 9. Springer Science & Business Media, 1989 (cit. on p. 2).
- [119]Luisa Peterson, Jens Bremer, and Kai Sundmacher. "Challenges in data-based reactor modeling: A critical analysis of purely data-driven and hybrid models for a CSTR case study". In: *Computers & Chemical Engineering* (2024), p. 108643 (cit. on p. 45).
- [120]Bruce E. Poling. *The properties of gases and liquids*. 2004 (cit. on pp. 3, 60, 61).
- [121]S. Prasanna and R. J. Doerksen. "Topological polar surface area: a useful descriptor in 2D-QSAR". In: *Current Medicinal Chemistry* 16.1 (2009), pp. 21–41 (cit. on p. 63).
- [122]John M. Prausnitz, Rudiger N. Lichtenthaler, and Edmundo Gomes De Azevedo. *Molecular thermodynamics of fluid-phase equilibria*. Pearson Education, 1998 (cit. on p. 16).
- [123]*PubChem*. <https://pubchem.ncbi.nlm.nih.gov>. Accessed: 01 March 2024 (cit. on p. 98).

- [124]Cory C. Pye, Tom Ziegler, Erik Van Lenthe, and Jaap N. Louwen. “An implementation of the conductor-like screening model of solvation within the Amsterdam density functional package—Part II. COSMO for real solvents 1.” In: *Canadian Journal of Chemistry* 87.7 (2009), pp. 790–797 (cit. on p. 24).
- [125]Shiyi Qin, Shengli Jiang, Jianping Li, Prasanna Balaprakash, Reid C. Van Lehn, and Victor M. Zavala. “Capturing molecular interactions in graph neural networks: a case study in multi-component phase equilibrium”. In: *Digital Discovery* 2.1 (2023), pp. 138–151 (cit. on pp. 25, 52–55, 64, 65, 71, 92, 117, 163).
- [126]Bachan S. Rawat, Amar N Goswami, and Sri Krishna. “Isobaric vapour-liquid equilibria of the ternary system hexane-benzene-sulpholane”. In: *Journal of Chemical Technology and Biotechnology* 30.1 (1980), pp. 557–562 (cit. on p. 110).
- [127]RDKit: *Open-source cheminformatics*. <http://www.rdkit.org>. Accessed: December 2023 (cit. on pp. 27, 81, 132).
- [128]Henri Renon and John M. Prausnitz. “Local compositions in thermodynamic excess functions for liquid mixtures”. In: *AIChE Journal* 14.1 (1968), pp. 135–144 (cit. on pp. 16, 18).
- [129]Jean-Louis Reymond and Mahendra Awale. “Exploring chemical space for drug discovery using the chemical universe database”. In: *ACS Chemical Neuroscience* 3.9 (2012), pp. 649–657 (cit. on p. 4).
- [130]Kenneth Ridgway and Paul Alexander Butler. “Physical properties of the ternary system benzene-cyclohexane-hexane”. In: *Journal of Chemical and Engineering Data* 12.4 (1967), pp. 509–515 (cit. on p. 110).
- [131]Bernard Riedl and Robert E. Prud’Homme. “Thermodynamic study of poly (vinyl chloride)/polyester blends by inverse-phase gas chromatography at 120 °C”. In: *Journal of Polymer Science Part B: Polymer Physics* 24.11 (1986), pp. 2565–2582 (cit. on p. 193).
- [132]Jan G. Rittig, Karim Ben Hicham, Artur M. Schweidtmann, Manuel Dahmen, and Alexander Mitsos. “Graph neural networks for temperature-dependent activity coefficient prediction of solutes in ionic liquids”. In: *Computers & Chemical Engineering* 171 (2023), p. 108153 (cit. on pp. 25, 71, 82, 83, 85, 107).
- [133]Kunal Roy, Supratik Kar, and Rudra Narayan Das. *A primer on QSAR/QSPR modeling: fundamental concepts*. Springer Cham, 2015, p. 121 (cit. on p. 3).
- [134]Yvonne H. Frentzen Rudolph P. M. Guit. “Recovery of ϵ -caprolactam”. US 6,191,274 B1. DSM N.V., E.I. du Pont Nemous, and Company. Feb. 2001 (cit. on p. 131).
- [135]Paul N. Rylander. *Solvent extraction of oil-soluble water-immiscible alcohols using dimethylsulfoxide*. US Patent 2,954,392. 1960 (cit. on p. 127).
- [136]Edgar Ivan Sanchez Medina, Sreekanth Kunchapu, and Kai Sundmacher. “Gibbs–Helmholtz Graph Neural Network for the prediction of activity coefficients of polymer solutions at infinite dilution”. In: *The Journal of Physical Chemistry A* 127.46 (2023), pp. 9863–9873 (cit. on p. 25).

- [137]Edgar Ivan Sanchez Medina, Steffen Linke, Martin Stoll, and Kai Sundmacher. “Gibbs-Helmholtz Graph Neural Network: capturing the temperature dependency of activity coefficients at infinite dilution”. In: *Digital Discovery* 2 (3 2023), pp. 781–798 (cit. on p. 25).
- [138]Edgar Ivan Sanchez Medina, Steffen Linke, Martin Stoll, and Kai Sundmacher. “Graph neural networks for the prediction of infinite dilution activity coefficients”. In: *Digital Discovery* 1 (3 2022), pp. 216–225 (cit. on p. 25).
- [139]Edgar Ivan Sanchez Medina and Kai Sundmacher. “Solvent pre-selection for extractive distillation using Gibbs-Helmholtz Graph Neural Networks”. In: *Computer Aided Chemical Engineering*. Vol. 52. Elsevier, 2023, pp. 2037–2042 (cit. on p. 17).
- [140]S. I. Sandler. “Infinite dilution activity coefficients in chemical, environmental and biochemical engineering”. In: *Fluid Phase Equilibria* 116.1-2 (1996), pp. 343–353 (cit. on p. 17).
- [141]Dante H. Sarno. *Extractive distillation with dimethylformamide*. US Patent 2,993,841. 1961 (cit. on p. 125).
- [142]Kenneth W. Saunders. “Nitriles as selective solvents”. In: *Industrial & Engineering Chemistry* 43.1 (1951), pp. 121–126 (cit. on p. 123).
- [143]Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. “The graph neural network model”. In: *IEEE transactions on neural networks* 20.1 (2008), pp. 61–80 (cit. on p. 21).
- [144]Christian S. Schacht, Lawien Zubeir, Theo W. de Loos, and Joachim Gross. “Application of infinite dilution activity coefficients for determining binary equation of state parameters”. In: *Industrial & Engineering Chemistry Research* 49.16 (2010), pp. 7646–7653 (cit. on p. 17).
- [145]Junior D. Seader, Ernest J. Henley, and D. Keith Roper. *Separation process principles*. Wiley New York, 2006 (cit. on pp. 1, 10).
- [146]John Shalf. “The future of computing beyond Moore’s Law”. In: *Philosophical Transactions of the Royal Society A* 378.2166 (2020), p. 20190061 (cit. on p. 4).
- [147]David S. Sholl and Ryan P. Lively. “Seven chemical separations to change the world”. In: *Nature* 532.7600 (2016), pp. 435–437 (cit. on p. 1).
- [148]Martin Simonovsky and Nikos Komodakis. “Dynamic edge-conditioned filters in convolutional neural networks on graphs”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 3693–3702 (cit. on p. 30).
- [149]B. V. Subbarao and C. Venkatarao. “Isobaric ternary vapor-liquid equilibria. System: acetone-carbon tetrachloride-benzene.” In: *Journal of Chemical and Engineering Data* 11.2 (1966), pp. 158–162 (cit. on p. 110).
- [150]Mukund Sundararajan, Ankur Taly, and Qiqi Yan. “Axiomatic attribution for deep networks”. In: *International Conference on Machine Learning*. PMLR. 2017, pp. 3319–3328 (cit. on p. 136).

- [151] Tian Tan, Hongye Cheng, Guzhong Chen, Zhen Song, and Zhiwen Qi. “Prediction of infinite-dilution activity coefficients with neural collaborative filtering”. In: *AIChE Journal* 68.9 (2022), e17789 (cit. on pp. 25, 50).
- [152] Lei Tao, Vikas Varshney, and Ying Li. “Benchmarking machine learning models for polymer informatics: an example of glass transition temperature”. In: *Journal of Chemical Information and Modeling* 61.11 (2021), pp. 5395–5413 (cit. on p. 88).
- [153] Eugene R. Thomas and Charles A. Eckert. “Prediction of limiting activity coefficients by a modified separation of cohesive energy density model and UNIFAC”. In: *Industrial & Engineering Chemistry Process Design and Development* 23.2 (1984), pp. 194–209 (cit. on pp. 24, 176).
- [154] Ulf Tilstam. “Sulfolane: A versatile dipolar aprotic solvent”. In: *Organic Process Research & Development* 16.7 (2012), pp. 1273–1278 (cit. on p. 123).
- [155] Raghunath P. Tripathi and Lionel Asselineau. “Isobaric vapor-liquid equilibria in ternary system benzene-n-heptane-acetonitrile from binary T_x measurements”. In: *Journal of Chemical and Engineering Data* 20.1 (1975), pp. 33–40 (cit. on p. 110).
- [156] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. “Graph attention networks”. In: *arXiv:1710.10903* (2017) (cit. on p. 21).
- [157] Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. “Order matters: Sequence to sequence for sets”. In: *arXiv:1511.06391* (2015) (cit. on p. 31).
- [158] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, et al. “SciPy 1.0: fundamental algorithms for scientific computing in Python”. In: *Nature Methods* 17.3 (2020), pp. 261–272 (cit. on pp. 105, 121).
- [159] Moritz Von Stosch, Rui Oliveira, Joana Peres, and Sebastião Feye de Azevedo. “Hybrid semi-parametric modeling in process systems engineering: Past, present and future”. In: *Computers & Chemical Engineering* 60 (2014), pp. 86–101 (cit. on p. 41).
- [160] Pavel Vrbka and Vladimír Dohnal. “Limiting activity coefficient measurements in binary mixtures of dichloromethane and 1-alkanols (C_1 – C_4)”. In: *Fluid Phase Equilibria* 411 (2016), pp. 59–65 (cit. on p. 49).
- [161] Ulrich Weidlich and Juergen Gmehling. “A modified UNIFAC model. 1. Prediction of VLE, h^E , and γ^∞ ”. In: *Industrial & Engineering Chemistry Research* 26.7 (1987), pp. 1372–1381 (cit. on pp. 24, 56, 178).
- [162] David Weininger. “SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules”. In: *Journal of Chemical Information and Computer Sciences* 28.1 (1988), pp. 31–36 (cit. on p. 27).
- [163] Oliver Wieder, Stefan Kohlbacher, Méline Kuenemann, Arthur Garon, Pierre Ducrot, Thomas Seidel, and Thierry Langer. “A compact review of molecular property prediction with graph neural networks”. In: *Drug Discovery Today: Technologies* 37 (2020), pp. 1–12 (cit. on p. 5).

- [164] Grant M. Wilson. “Vapor-liquid equilibrium. XI. A new expression for the excess free energy of mixing”. In: *Journal of the American Chemical Society* 86.2 (1964), pp. 127–130 (cit. on p. 16).
- [165] Benedikt Winter, Clemens Winter, Timm Esper, Johannes Schilling, and André Bardow. “SPT-NRTL: A physics-guided machine learning model to predict thermodynamically consistent activity coefficients”. In: *Fluid Phase Equilibria* 568 (2023), p. 113731 (cit. on pp. 25, 107).
- [166] Benedikt Winter, Clemens Winter, Johannes Schilling, and André Bardow. “A smile is all you need: predicting limiting activity coefficients from SMILES with natural language processing”. In: *Digital Discovery* 1.6 (2022), pp. 859–869 (cit. on pp. 25, 50, 72–74, 79).
- [167] Lingfei Wu, Peng Cui, Jian Pei, Liang Zhao, and Xiaojie Guo. *Graph neural networks: foundation, frontiers and applications*. Springer Nature Singapore, 2022, p. 689 (cit. on p. 20).
- [168] Gong Xingchu, Lü Yangcheng, Li Mu, Ma Xuefeng, Ni Qianyin, and Luo Guangsheng. “Selection and evaluation of a new extractant for caprolactam extraction”. In: *Chinese Journal of Chemical Engineering* 16.6 (2008), pp. 876–880 (cit. on p. 131).
- [169] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. “Representation learning on graphs with jumping knowledge networks”. In: *International Conference on Machine Learning*. PMLR, 2018, pp. 5453–5462 (cit. on p. 31).
- [170] Shiqiang Zhang, Juan S. Campos, Christian Feldmann, Frederik Sandfort, Miriam Mathea, and Ruth Misener. “Augmenting optimization-based molecular design with graph neural networks”. In: *Computers & Chemical Engineering* (2024), p. 108684 (cit. on p. 139).
- [171] Chongli Zhong, Yoshiyuki Sato, Hirokatsu Masuoka, and Xiaoning Chen. “Improvement of predictive accuracy of the UNIFAC model for vapor-liquid equilibria of polymer solutions”. In: *Fluid Phase Equilibria* 123.1-2 (1996), pp. 97–106 (cit. on pp. 87, 88, 91–93).
- [172] Teng Zhou, Zhen Song, Xiang Zhang, Rafiqul Gani, and Kai Sundmacher. “Optimal solvent design for extractive distillation processes: a multiobjective optimization-based hierarchical framework”. In: *Industrial & Engineering Chemistry Research* 58.15 (2019), pp. 5777–5786 (cit. on p. 127).

List of Figures

1.1	Depiction of the main techniques used for separation processes design throughout history.	2
2.1	Main hybrid semi-parametric model arrangements.	18
3.1	Solute-solvent matrix representation of the available IDAC data at 298.15 K on the <i>Brouwer data set</i> . Available IDAC data is represented by colored entries.	26
3.2	Schematic representation outlining the process of generating molecular graphs from SMILES strings.	28
3.3	Schematic representation of the message-passing layer used for the prediction of isothermal IDACs.	30
3.4	Schematic representation of the proposed GNN-based model for the prediction of isothermal IDACs.	33
3.5	Incremental performance in the mean absolute percentage error (MAPE) with respect to the ensemble size of the proposed GNN-based for isothermal IDACs.	35
3.6	Parity plot between the experimental and the predicted IDAC values by GNN ensemble and UNIFAC-Dortmund. All the feasible systems for each method are shown.	37
3.7	Mean absolute percentage error (MAPE) for the proposed GNN and popular phenomenological models when predicting IDACs at 298.15 K. Results are shown only for feasible systems of the corresponding phenomenological model contained in the GNN test set during a 5-fold cross-validation.	40
3.8	Absolute error density of UNIFAC (Lyngby), the GNN ensemble and the corresponding hybrid GNN model. The black central line shows the null-error for comparison. The results are shown for systems in the test set.	43
3.9	Absolute error density of UNIFAC, the GNN ensemble and the corresponding hybrid GNN model. The black central line shows the null-error for comparison. The results are shown for systems in the test set.	44

3.10	Absolute error density of UNIFAC (Dortmund), the GNN ensemble and the corresponding hybrid GNN model. The black central line shows the null-error for comparison. The results are shown for systems in the test set.	45
3.11	Absolute error density of COSMO-RS the GNN ensemble and the corresponding hybrid GNN model. The black central line shows the null-error for comparison. The results are shown for systems in the test set. . . .	46
3.12	Absolute error density of Abraham, the GNN ensemble and the corresponding hybrid GNN model. The black central line shows the null-error for comparison. The results are shown for systems in the test set. . . .	47
3.13	Absolute error density of MOSCED, the GNN ensemble and the corresponding hybrid GNN model. The black central line shows the null-error for comparison. The results are shown for systems in the test set. . . .	47
3.14	Percentage of systems predicted within the absolute error thresholds 0.1, 0.2 and 0.3. Values correspond to systems in the test set of the corresponding isothermal subset that are feasible to UNIFAC-Dortmund.	56
4.1	Schematic illustration of a two-component mixture graph.	64
4.2	Schematic illustration of the molecular GNN in the Gibbs-Helmholtz Graph Neural Network.	66
4.3	Schematic illustration of the mixture GNN in the Gibbs-Helmholtz Graph Neural Network.	69
4.4	Parity plot between the experimental and the predicted IDAC values by the GH-GNN while interpolating to other temperatures.	75
4.5	Parity plot between the experimental and the predicted IDAC values by the GH-GNN while extrapolating to temperature T_{high}	76
4.6	Parity plot between the experimental and the predicted IDAC values by the GH-GNN while extrapolating to temperature T_{low}	76
4.7	Parity plot between the experimental and the predicted IDAC values by the GH-GNN and the UNIFAC-Dortmund models while interpolating among binary-systems. The results for UNIFAC-Dortmund are shown for all feasible systems in the test set excluding the worst 9 predictions.	78
4.8	Parity plot between the experimental and the predicted IDAC values by the GH-GNN model on the external data set.	80
4.9	Progression of the mean absolute error (MAE) achieved by the GH-GNN model on systems in the external data set that fall into different Jaccard distance thresholds.	81

4.10	Cumulative distribution of the absolute prediction error of the extended GH-GNN and the UNIFAC-IL [32] models. The results are shown for the test set.	84
4.11	Graph polymer representations used in the extension of the GH-GNN model to polymer solutions.	88
4.12	Mean absolute error (MAE) achieved by the extended GH-GNN trained with different polymer representations on each subset.	89
5.1	Schematic illustration of the proposed hybrid serial model consisting of the GH-GNN model and the extended Margules equation for predicting activity coefficients.	97
5.2	Heatmap of the mean absolute error achieved by the GH-GNN-Margules model on all KDB isothermal data according to binary chemical classes. The number on each cell indicates the number of data points in each subset.	101
5.3	Cumulative percentage of binary classes predicted by the GH-GNN-Margules and UNIFAC-Dortmund models within different mean absolute error thresholds. The errors are calculated according to the predicted molar fraction in the vapor phase on the isothermal vapor-liquid equilibria KDB data. Only feasible systems for UNIFAC-Dortmund are considered.	102
5.4	Isothermal vapor-liquid equilibria diagram of two systems that could not be predicted with UNIFAC-Dortmund, but were predicted by GH-GNN-Margules. Markers denote experimental measurements.	103
5.5	Heatmap of the mean absolute error achieved by the GH-GNN-Margules model on all KDB isobaric data according to binary chemical classes. The number on each cell indicates the number of data points in each subset.	106
5.6	Isobaric vapor-liquid equilibria diagram of system "pyridine/1,2,3,4-tetrahydronaphthalene" at 26.66 kPa predicted with GH-GNN-Margules and UNIFAC-Dortmund.	107
5.7	Isobaric vapor-liquid equilibria diagram of system "tetrachloroethylene/furfural" at 101.325 kPa predicted with GH-GNN-Margules and UNIFAC-Dortmund.	108
5.8	Matrix of binary classes contained in the KDB data set that are feasible to predict with UNIFAC-Dortmund. The color shows whether UNIFAC-Dortmund or GH-GNN-Margules achieve a lower mean absolute error (MAE). The number on each cell indicates the number of data points in each subset.	109

5.9	Ternary vapor-liquid equilibria for the system “chloroform/methanol/benzene” at 101.325 kPa. Experimental tie lines are taken from [92].	111
5.10	Ternary vapor-liquid equilibria for the system “hexane/methanol/acetone” at 313.15 K. Experimental tie lines are taken from [112].	112
6.1	Effect on the relative volatility of the mixture benzene/heptane caused by different solvent-to-feed ratios (SF) of the top 5 solvents identified in the pre-selection process.	123
6.2	Phase equilibrium behavior of the ternary system [EMIM][BF ₄], caprolactam and the indicated solvent as predicted by COSMO-RS, and measured experimentally. Adapted from [105].	130
6.3	Comparison of solvent rankings according to the extended GH-GNN model and UNIFAC-IL [32] for the extraction case-study of [EMIM][BF ₄] and caprolactam.	131
6.4	Parity comparison between the predicted logarithmic lignin solubility in terms of its mole fraction given by COSMO-RS and the GNN.	134
6.5	Correspondence of the GNN for classification and the GNN for regression for the logarithmic lignin solubility in terms of its mole fraction as predicted by COSMO-RS.	137
6.6	Depiction of the attribution scores (normalized to 0-1) for diethyl sulfoxide (left), thiazole (middle) and 4-methyl-pyrimidine (right). The magnitude of the attribution score is depicted proportional to the darkness of the color.	138
A.1	Proportion of data points contained in the train and test data sets, as well as in the entire data set.	183
A.2	Number of data points used in the construction of the DECHEMA data set according to the experimental technique.	186
A.3	Distribution of logarithmic IDACs in the DECHEMA data set.	187
A.4	Distribution of temperature values in the DECHEMA data set.	188
A.5	Number of compounds in the DECHEMA data set contained in the 30 most frequent chemical classes computed by Classyfire [38].	189

List of Tables

3.1	Atomic features defining the initial feature vector of nodes in the molecular graphs constructed from the Brouwer data set. The dimension of the corresponding one-hot encoded feature is also shown.	28
3.2	Bond features defining the initial feature vector of edges in the molecular graphs constructed from the Brouwer data set. The dimension of the corresponding one-hot encoded feature is also shown.	28
3.3	Performance comparison between the proposed GNN model and popular phenomenological models for isothermal IDAC prediction.	36
3.4	Performance comparison between the proposed GNN model and popular phenomenological models using 5-fold cross validation when predicting IDACs at 298.15 K.	40
3.5	Performance comparison between the corresponding hybrid parallel GNN, the GNN ensemble (e-GNN) and the phenomenological models. Results are shown for the systems in the test set. The best value for each method is shown in bold.	42
3.6	Comparison of the DECHEMA data set against similar IDAC data sets used in the literature.	50
3.7	Information of the isothermal subsets obtained from the DECHEMA data set and used for the extended isothermal study.	51
3.8	Comparison performance of UNIFAC-Dortmund, MOSCED, COSMO-RS, the proposed e-GNN and e-SolvGNN [125] models on predicting $\ln \gamma_i^\infty$ on various isothermal subsets.	53
4.1	Atomic features defining the initial feature vector of nodes in the molecular graphs constructed from the DECHEMA data set. The dimension of the corresponding one-hot encoded feature is also shown.	62
4.2	Bond features defining the initial feature vector of edges in the molecular graphs constructed from the DECHEMA data set. The dimension of the corresponding one-hot encoded feature is also shown.	62

4.3	Global features defining the initial feature vector of molecular graphs constructed from the DECHEMA data set and used in the temperature-dependent IDAC studies. The dimension of the corresponding one-hot encoded feature is also shown.	63
4.4	Performance comparison of GNN-based models and UNIFAC-Dortmund for the prediction of temperature-dependent IDACs.	72
4.5	Performance of the GH-GNN and GNNCat models for predicting IDACs while interpolating among temperature values, extrapolating to lower temperatures and extrapolating to higher temperatures.	75
4.6	Performance of the GH-GNN and UNIFAC-Dortmund models for predicting IDACs while interpolating among binary-systems. Results excluding the worst 9 predictions are indicated by “w/o”.	78
4.7	Performance of the GH-GNN and UNIFAC-Dortmund models for predicting IDACs while extrapolating to other solutes or solvents. Results excluding the worst 9 predictions are indicated by “w/o”.	79
4.8	Performance of different models for predicting IDACs of systems containing ionic liquids (ILs).	83
4.9	Information of the three distinct data subsets used for the extension of the GH-GNN model to polymer solutions.	87
4.10	Interpolation performance achieved by the extended GH-GNN model (with and without pre-training on organic systems) and the random forest baseline using the periodic unit representation. The standard deviation is shown in between parenthesis.	90
4.11	Extrapolation performance achieved by the extended GH-GNN model (with and without pre-training on organic systems) and the random forest baseline using the periodic unit representation. The standard deviation is shown in between parenthesis.	91
4.12	Comparison between the phenomenological models UNIFAC-ZM [171] and Entropic-FV [90] against the GH-GNN model extended for polymer solutions.	92
5.1	General information of the KDB data set consisting of binary VLE experimental measurements.	99
5.2	GH-GNN-Margules and UNIFAC-Dortmund performance on predicting isothermal binary vapor-liquid equilibria data. The metrics are shown with respect to the predicted molar fraction in the vapor phase.	102
5.3	GH-GNN-Margules and UNIFAC-Dortmund performance on predicting isobaric binary vapor-liquid equilibria data. The metrics are shown with respect to the predicted molar fraction in the vapor phase.	106

5.4	GH-GNN-Margules and UNIFAC-Dortmund performance on predicting all binary vapor-liquid equilibria data contained in the KDB data set. The metrics are shown with respect to the predicted molar fraction in the vapor phase.	108
5.5	Ternary vapor-liquid equilibria data used for evaluating the performance of the GH-GNN-Margules model.	110
5.6	GH-GNN-Margules and UNIFAC-Dortmund performance on predicting ternary vapor-liquid equilibria. The metrics are shown with respect to the predicted molar fractions of components 1 and 2 in the vapor phase.	112
6.1	Top 5 solvents selected for the indicated aromatic/aliphatic mixtures. The number between parenthesis indicates the value for the corresponding metric.	122
6.2	Top 5 solvents selected for the indicated paraffin/olefin mixtures. The number between parenthesis indicates the value for the corresponding metric.	125
6.3	Top 5 solvents selected for the indicated mixtures containing oxygenated compounds. The number between parenthesis indicates the value for the corresponding metric.	126
6.4	Top 10 solvents selected for the liquid-liquid extraction of caprolactam from [EMIM][BF ₄] after considering the normal boiling point, molecular weight and GHS classification filters. The raking metric used corresponds to Eq. 6.10. The conventional solvent is marked with *.	129
6.5	Atomic features defining the initial feature vector of nodes in the molecular graphs constructed from the lignin solubility COSMO-RS calculations. The dimension of the corresponding one-hot encoded feature is also shown.	133
6.6	Bond features defining the initial feature vector of edges in the molecular graphs constructed from the lignin solubility COSMO-RS calculations. The dimension of the corresponding one-hot encoded feature is also shown.	133
A.1	Fixed hyperparameters based on experience.	182
A.2	Tuned hyperparameters using Bayesian optimization, exploration bounds and final selected values.	182
A.3	Meaning of the experimental technique identifiers.	187
A.4	Compounds in the DECHEMA data set that were not assigned to a specific chemical class using Classyfire [38].	188
A.5	Ranges used during the hyperparameter search of e-GNNprev in the isothermal studies.	190

A.6	Final selected hyperparameters for e-GNNprev in each isothermal study.	190
A.7	Ranges used during the hyperparameter search of e-SolvGNN in the isothermal studies.	191
A.8	Final selected hyperparameters for e-SolvGNN in each isothermal study.	191
A.9	Hyperparameter details for GNNCat.	192
A.10	Hyperparameter details for SolvGNNCat.	192

Appendices

A.1 Relationship between partial molar properties and state variables

In order to show the relationship between a state variable and the corresponding partial molar properties, one has to rely on the Euler's theorem in the context of differential geometry. This theorem states that for any homogeneous function of degree k , the following is true

$$x_1 \frac{\partial f}{\partial x_1} + x_2 \frac{\partial f}{\partial x_2} + \cdots + x_n \frac{\partial f}{\partial x_n} = k f(x_1, x_2, \dots, x_n) \quad (\text{A.1})$$

A homogeneous function of degree k is such that the following relationship is true

$$f(\lambda x_1, \lambda x_2, \dots, \lambda x_n) = \lambda^k f(x_1, x_2, \dots, x_n) \quad (\text{A.2})$$

this is also true for the special case of $k = 1$, such that

$$f(\lambda x_1, \lambda x_2, \dots, \lambda x_n) = \lambda f(x_1, x_2, \dots, x_n) \quad (\text{A.3})$$

in which case we say that f is a homogeneous function of degree 1. Any extensive state thermodynamic variable (e.g., mass, energy, volume) of a mixture is a homogeneous function of degree 1. The intuition behind this is that if one maintains everything constant and just doubles the amount of each component n_i in the mixture, the extensive variable will double in value. Therefore, for any extensive state variable, we can write

$$M(T, P, \lambda n_1, \lambda n_2, \dots, \lambda n_n) = \lambda M(T, P, n_1, n_2, \dots, n_n) \quad (\text{A.4})$$

If we differentiate Eq. A.4 with respect to λ we obtain that

$$\left. \frac{\partial M}{\partial(\lambda n_1)} \frac{\partial(\lambda n_1)}{\partial \lambda} \right|_{T,P} + \left. \frac{\partial M}{\partial(\lambda n_2)} \frac{\partial(\lambda n_2)}{\partial \lambda} \right|_{T,P} + \dots + \left. \frac{\partial M}{\partial(\lambda n_n)} \frac{\partial(\lambda n_n)}{\partial \lambda} \right|_{T,P} = M(T, P, n_1, n_2, \dots, n_n) \quad (\text{A.5})$$

which is exactly the relationship between the state variable M and the corresponding partial molar properties:

$$M = \sum_i n_i \bar{m}_i \quad (\text{A.6})$$

or in terms of molar fractions

$$m = \sum_i z_i \bar{m}_i \quad (\text{A.7})$$

A.2 Chemical potential in terms of fugacity

Starting from fundamental equation of the Gibbs energy for a N -component system with material exchange

$$dG = -SdT + VdP + \sum_i^N \mu_i dn_i \quad (\text{A.8})$$

one can compare it to the total differential of G

$$dG = \left(\frac{\partial G}{\partial T}\right)_{P,\mathbf{n}} dT + \left(\frac{\partial G}{\partial P}\right)_{T,\mathbf{n}} dP + \sum_i \left(\frac{\partial G}{\partial n_i}\right)_{T,P,\mathbf{n}_{\neq i}} dn_i \quad (\text{A.9})$$

where $\mathbf{n} = [n_1, n_2, \dots, n_N]$ is the vector of number of mols for all components in the mixture, and $\mathbf{n}_{\neq i}$ stands for the vector of number of mols for all components except component i ; and then note that

$$S = -\left(\frac{\partial G}{\partial T}\right)_{P,\mathbf{n}} \quad (\text{A.10})$$

$$V = \left(\frac{\partial G}{\partial P}\right)_{T,\mathbf{n}} \quad (\text{A.11})$$

$$\mu_i = \left(\frac{\partial G}{\partial n_i}\right)_{T,P,\mathbf{n}_{\neq i}} \quad (\text{A.12})$$

Given these primary relationships, and by applying the Schwarz theorem of differential calculus, one can also obtain that

$$-\left(\frac{\partial S}{\partial P}\right)_{T,\mathbf{n}} = \left(\frac{\partial V}{\partial T}\right)_{P,\mathbf{n}} \quad (\text{A.13})$$

$$-\left(\frac{\partial S}{\partial n_i}\right)_{T,P,\mathbf{n}_{\neq i}} = \left(\frac{\partial \mu_i}{\partial T}\right)_{P,\mathbf{n}} \quad (\text{A.14})$$

$$\left(\frac{\partial V}{\partial n_i}\right)_{T,P,\mathbf{n}_{\neq i}} = \left(\frac{\partial \mu_i}{\partial P}\right)_{T,\mathbf{n}} \quad (\text{A.15})$$

These relationships, known as the Maxwell relationships, apply to any multi-component system with material exchange despite its state. Therefore, they also apply for the ideal gas. In particular, if we consider the right-hand side of Eq. A.15 at the conditions of constant temperature T and constant number of mols \mathbf{n} , the derivative is in fact ordinary. Then, by using the ideal gas equation for obtaining the derivative of the left-hand side of Eq. A.15 we obtain that

$$d\mu_i^{id} = \frac{RT}{P} dP \quad (\text{A.16})$$

which we can integrate from a reference pressure P^0 to the system's pressure P at constant T and \mathbf{z} to obtain

$$\mu_i^{id}(T, P, \mathbf{z}) = \mu_i^{id}(T, P^0, \mathbf{z}) + RT \ln \frac{P}{P^0} \quad (\text{A.17})$$

where \mathbf{z} is the vector of molar fractions of all mixture components.

In practice, Eq. A.17 would require the knowledge of the reference chemical potential at many different compositions (depending on the composition of the actual mixture). Therefore, this expression is not very practical or, better to say, this reference state is not very practical. Instead, one might choose the reference state of the pure component i as ideal gas where the following is true

$$\mu_i^{id}(T, P^0, \mathbf{z}) = \mu_i^{id,pure}(T, p) \quad (\text{A.18})$$

notice that, from the ideal gas equation, p corresponds to the partial pressure of component i in the system that is at pressure P^0 (i.e., $p = z_i P^0$). And, by using the same reference pressure P^0 for the pure ideal gas we obtain that

$$\mu_i^{id}(T, P^0, \mathbf{z}) = \mu_i^{id,pure}(T, z_i P^0) = \mu_i^{id,pure}(T, P^0) + RT \ln z_i \quad (\text{A.19})$$

Therefore, Eq. A.17 can be written in terms of a more practical reference point as

$$\mu_i^{id}(T, P, \mathbf{z}) = \mu_i^{id,pure}(T, P^0) + RT \ln \frac{z_i P}{P^0} \quad (\text{A.20})$$

One idea for extending Eq. A.20 from ideal gases to real mixtures was proposed by Lewis via the introduction of an auxiliary variable called *fugacity*. This auxiliary

variable serves as a modified partial pressure, accounting for deviations from ideal behavior. The general expression can then be written as

$$\mu_i(T, P, \mathbf{z}) = \mu_i^{pure}(T, P^0) + RT \ln \frac{f_i(T, P, \mathbf{z})}{f_i^{0,pure}(T, P^0)} \quad (\text{A.21})$$

A.3 Derivation of the dependency expressions of γ_i on T , P and x

The differential expression relating γ_i to the mixture temperature T can be obtained from the fundamental expression of the molar Gibbs energy in terms of the excess properties

$$g^E = h^E - Ts^E = u^E + Pv^E - Ts^E \quad (\text{A.22})$$

dividing both sides by RT , and taking the partial derivative with respect to T

$$\left. \frac{\partial(g^E/RT)}{\partial T} \right|_P = \frac{RT \left. \frac{\partial h^E}{\partial T} \right|_P - RTs^E - RT^2 \left. \frac{\partial s^E}{\partial T} \right|_P - Rh^E + RTs^E}{(RT)^2} \quad (\text{A.23})$$

simplifying

$$\left. \frac{\partial(g^E/RT)}{\partial T} \right|_P = \frac{1}{RT} \left. \frac{\partial h^E}{\partial T} \right|_P - \frac{1}{R} \left. \frac{\partial s^E}{\partial T} \right|_P - \frac{h^E}{RT^2} \quad (\text{A.24})$$

and from the fundamental equation of h^E and the total differentials of h^E and s^E we know that

$$\left. \frac{\partial h^E}{\partial T} \right|_P = T \left. \frac{\partial s^E}{\partial T} \right|_P \quad (\text{A.25})$$

therefore,

$$\left. \frac{\partial(g^E/RT)}{\partial T} \right|_P = -\frac{h^E}{RT^2} \quad (\text{A.26})$$

If we now consider the multi-component case, and introducing Eq. 2.29, we know that the previous relationship also holds for the partial molar properties of each individual component i

$$\left. \frac{\partial(\bar{g}_i^E/RT)}{\partial T} \right|_{P,x} = -\frac{\bar{h}_i^E}{RT^2} = \left. \frac{\partial(\ln \gamma_i)}{\partial T} \right|_{P,x} \quad (\text{A.27})$$

and now, by using a change of variable $u = 1/T$ and the chain rule, we obtain that

$$\boxed{\left. \frac{\partial \ln \gamma_i}{\partial(1/T)} \right|_{P,\mathbf{x}} = \frac{\bar{h}_i^E}{R}} \quad (\text{A.28})$$

Similarly, the expression for the pressure dependency can be derived by taking the derivative of Eq. A.22 with respect to the system's pressure P

$$\left. \frac{\partial g^E}{\partial P} \right|_T = \left. \frac{\partial u^E}{\partial P} \right|_T + v^E + P \left. \frac{\partial v^E}{\partial P} \right|_T - T \left. \frac{\partial s^E}{\partial P} \right|_T \quad (\text{A.29})$$

and by using the fundamental equation of the molar excess internal energy

$$Pdv^E = du^E - Tds^E \quad (\text{A.30})$$

and taking the change with respect to pressure at constant T , we have that

$$P \left. \frac{\partial v^E}{\partial P} \right|_T = \left. \frac{\partial u^E}{\partial P} \right|_T - T \left. \frac{\partial s^E}{\partial P} \right|_T \quad (\text{A.31})$$

hence, we have that

$$\left. \frac{\partial g^E}{\partial P} \right|_T = v_i^E \quad (\text{A.32})$$

and, in terms of the partial molar properties

$$\left. \frac{\partial \bar{g}_i^E}{\partial P} \right|_{T,\mathbf{x}} = \bar{v}_i^E \quad (\text{A.33})$$

We can then introduce Eq. 2.28 to obtain the final expression

$$\boxed{\left. \frac{\partial \ln \gamma_i}{\partial P} \right|_{T,\mathbf{x}} = \frac{\bar{v}_i^E}{RT}} \quad (\text{A.34})$$

Finally, the dependency of γ_i on the mixture's composition can be obtained from the fundamental equation of the partial excess molar Gibbs energy

$$s^E dT - v^E dP + \sum_i^N x_i d\bar{g}_i^E = 0 \quad (\text{A.35})$$

at constant T and P

$$\sum_i^N x_i d\bar{g}_i^E = 0 \quad (\text{A.36})$$

and introducing Eq. 2.28 we obtain the final expression

$$\boxed{\sum_i^N x_i d \ln \gamma_i = 0} \quad (\text{A.37})$$

A.4 Derivation of the relationship of excess chemical potential and γ_i

The relationship between the excess chemical potential and the activity coefficient is obtained by subtracting the chemical potential in the ideal solution *ids* (which is analogous to Eq. A.20 for the ideal gas)

$$\mu_i^{ids}(T, P, \mathbf{x}) = \mu_i^{pure}(T, P^0) + RT \ln \frac{x_i f_i^{pure}(T, P)}{f_i^{0,pure}(T, P^0)} \quad (\text{A.38})$$

from the chemical potential of the real mixture (Eq. 2.13) to obtain

$$\bar{g}_i^E(T, P, \mathbf{x}) = RT \left(\ln \frac{f_i(T, P, \mathbf{x})}{f_i^{0,pure}(T, P^0)} - \ln \frac{x_i f_i^{pure}(T, P)}{f_i^{0,pure}(T, P^0)} \right) \quad (\text{A.39})$$

which simplifies to

$$\bar{g}_i^E(T, P, \mathbf{x}) = RT \ln \frac{f_i(T, P, \mathbf{x})}{x_i f_i^{pure}(T, P)} \quad (\text{A.40})$$

The ratio on the right-hand side of Eq. A.40 correspond to the activity coefficient (cf. Eq. 2.18). Therefore, we arrive to the final expression

$$\bar{g}_i^E(T, P, \mathbf{x}) = RT \ln \gamma_i(T, P, \mathbf{x}) \quad (\text{A.41})$$

A.5 Solvation models: Hildebrand parameter, Hansen Solubility Parameters and MOSCED

Here, the mathematical formulations of the models for predicting infinite dilution activity coefficients (IDACs) that use the Hildebrand parameter [71] and the Hansen solubility parameters [65] are presented. Moreover, the formulation of the Modified Separation of Cohesive Energy Density (MOSCED) [153] model is introduced. All these three models rely on the entropic and enthalpic contributions to non-ideality, which, from the fundamental equation of the molar excess Gibbs energy, are additive. These terms have been historically referred to as the combinatorial and residual terms, for the entropic and enthalpic contributions, respectively.

$$\ln \gamma_i^\infty = \ln \gamma_i^{\infty,C} + \ln \gamma_i^{\infty,R} \quad (\text{A.42})$$

Hildebrand model:

$$\ln \gamma_i^{\infty,C} = \ln \frac{V_{m,j}}{V_{m,i}} + 1 - \frac{V_{m,j}}{V_{m,i}} \quad (\text{A.43})$$

$$\ln \gamma_i^{\infty,R} = \chi_{ij} \Phi_i^2 \quad (\text{A.44})$$

$$\chi_{ij} = \frac{V_{m,j}}{RT} (\delta_i - \delta_j)^2 \quad (\text{A.45})$$

$$\delta = \sqrt{c} = \sqrt{\frac{\Delta H_{vap} - RT}{V_m}} \quad (\text{A.46})$$

where, $V_{m,j}$ and $V_{m,i}$ are the molar volume of the solvent j and solute i respectively, R is the gas constant, T is the temperature, c is the cohesive energy density proposed by Hildebrand and ΔH_{vap} is the enthalpy of vaporization.

Hansen model: This model attempts to improve the predictions of the Hildebrand model by separating the solubility parameter δ into its contributions due to dispersion δ_D , polarity δ_P and hydrogen-bonding δ_{HB} , which are known as the Hansen Solubility Parameters (HSP).

$$\delta^2 = \delta_D^2 + \delta_P^2 + \delta_{HB}^2 \quad (\text{A.47})$$

MOSCED model:

$$\ln \gamma_i^{\infty,C} = \ln \left(\frac{V_{m,j}}{V_{m,i}} \right)^A + 1 - \left(\frac{V_{m,j}}{V_{m,i}} \right)^A \quad (\text{A.48})$$

$$\ln \gamma_i^{\infty,R} = \frac{V_{m,i}}{RT} \left((\lambda_j - \lambda_i)^2 + \frac{q_j^2 q_i^2 (\tau_j^T - \tau_i^T)^2}{\psi_j} + \frac{(\alpha_j^T - \alpha_i^T)(\beta_j^T - \beta_i^T)}{\xi_j} \right) \quad (\text{A.49})$$

$$A = 0.953 - 0.002314 \left((\tau_i^T)^2 + \alpha_i^T \beta_i^T \right) \quad (\text{A.50})$$

$$\alpha^T = \alpha \left(\frac{293}{T} \right)^{0.8} \quad (\text{A.51})$$

$$\beta^T = \beta \left(\frac{293}{T} \right)^{0.8} \quad (\text{A.52})$$

$$\tau^T = \tau \left(\frac{293}{T} \right)^{0.4} \quad (\text{A.53})$$

$$\psi_j = POL + 0.002629 \alpha_j^T \beta_j^T \quad (\text{A.54})$$

$$\xi_j = 0.68(POL - 1) + (3.4 - 2.4 \exp(-0.002687(\alpha_j \beta_j)^{1.5})) \left(\frac{293}{T} \right)^2 \quad (\text{A.55})$$

$$POL = q_j^4 (1.15 - 1.15 \exp(-0.002337(\tau_j^T)^3)) + 1 \quad (\text{A.56})$$

where τ is the polarity parameter, α and β are the hydrogen-bond acidity and basicity parameters, λ is the dispersion parameter and q is the induction parameter.

A.6 Group contribution models: UNIFAC, UNIFAC-Lyngby and UNIFAC-Dortmund

Here, the mathematical formulations of the UNIFAC [47], UNIFAC-Lyngby [95, 81] and UNIFAC-Dortmund [161, 57, 58, 59] models for predicting infinite dilution activity coefficients (IDACs) are presented. All these three models rely on the entropic and enthalpic contributions to non-ideality, which, from the fundamental equation of the molar excess Gibbs energy, are additive. These terms have been historically referred to as the combinatorial and residual terms, for the entropic and enthalpic contributions, respectively.

$$\ln \gamma_i^\infty = \ln \gamma_i^{\infty,C} + \ln \gamma_i^{\infty,R} \quad (\text{A.57})$$

Each group can have multiple subgroups.

UNIFAC model:

$$\ln \gamma_i^{\infty,C} = 1 - V_i + \ln V_i - 5q_i \left(1 - \frac{V_i}{F_i} + \ln \frac{V_i}{F_i} \right) \quad (\text{A.58})$$

$$V_i = \frac{r_i}{r_i x_i + r_j x_j} \quad (\text{A.59})$$

$$F_i = \frac{q_i}{q_i x_i + q_j x_j} \quad (\text{A.60})$$

$$r_i = \sum_k \nu_k^{(i)} R_k \quad (\text{A.61})$$

$$q_i = \sum_k \nu_k^{(i)} Q_k \quad (\text{A.62})$$

where, R_k and Q_k are the relative van der Waals volume and surface area for subgroup k , respectively. $\nu_k^{(i)}$ is the number of subgroups k in molecule i . Therefore, for computing the combinatorial part, the fragmentation of the involved molecules into subgroups needs to be feasible, and the parameters R_k and Q_k for each subgroup need to be available. The residual part is calculated as

$$\ln \gamma_i^{\infty, R} = \sum_k \nu_k^{(i)} (\ln \Gamma_k^\infty - \ln \Gamma_k^{\infty, (i)}) \quad (\text{A.63})$$

$$\ln \Gamma_k^\infty = Q_k \left[1 - \ln \left(\sum_m \Theta_m \Psi_{mk} \right) - \sum_m \frac{\Theta_m \Psi_{km}}{\sum_n \Theta_n \Psi_{nm}} \right] \quad (\text{A.64})$$

$$\Theta_m = \frac{Q_m X_m}{\sum_n Q_n X_n} \quad (\text{A.65})$$

$$X_m = \frac{\nu_m^{(i)} x_i + \nu_m^{(j)} x_j}{\sum_n \nu_n^{(i)} x_i + \sum_n \nu_n^{(j)} x_j} \quad (\text{A.66})$$

$$\ln \Gamma_k^{\infty, (i)} = Q_k \left[1 - \ln \left(\sum_m \Theta_m^{(i)} \Psi_{mk} \right) - \sum_m \frac{\Theta_m^{(i)} \Psi_{km}}{\sum_n \Theta_n^{(i)} \Psi_{nm}} \right] \quad (\text{A.67})$$

$$\Theta_m^{(i)} = \frac{Q_m X_m^{(i)}}{\sum_n Q_n X_n^{(i)}} \quad (\text{A.68})$$

$$X_m^{(i)} = \frac{\nu_m^{(i)} x_i}{\sum_n \nu_n^{(i)} x_i} \quad (\text{A.69})$$

$$\Psi_{nm} = \exp \left(-\frac{a_{nm}}{T} \right) \quad (\text{A.70})$$

where a_{nm} is the binary interaction parameter between group n and group m . By definition, $a_{nm} = 0$ if $n = m$, and $a_{nm} \neq a_{mn}$. T is the temperature. X_m is the mole fraction of subgroup m in the mixture, and Θ_m is the surface area fraction of subgroup m in the mixture. Γ_k^∞ stands for the infinite dilution activity coefficient of subgroup k in the mixture (i.e, the summation terms are over all mixture subgroups), and $\Gamma_k^{\infty, (i)}$ stands for the infinite dilution activity coefficient of subgroup k in the pure compound (i.e, the summation terms are over all subgroups in molecule i).

UNIFAC-Lyngby model: The residual part is calculated as in the original UNIFAC model. However, the binary interaction parameter a_{nm} is now made temperature dependent. The combinatorial part removes the Staverman-Guggenheim correction term and uses a (empirically determined) modified volume fraction term V_i' as follows

$$\ln \gamma_i^{\infty, C} = 1 - V_i' + \ln V_i' \quad (\text{A.71})$$

$$V_i' = \frac{r_i^{2/3}}{r_i^{2/3} x_i + r_j^{2/3} x_j} \quad (\text{A.72})$$

UNIFAC-Dortmund model: The residual part is calculated as in the original UNIFAC model. However, the binary interaction parameter a_{nm} is now made temperature dependent. The combinatorial part retains the Staverman-Guggenheim correction to a now empirically modified Flory-Huggins term

$$\ln \gamma_i^{\infty, C} = 1 - V_i'' + \ln V_i'' - 5q_i \left(1 - \frac{V_i}{F_i} + \ln \frac{V_i}{F_i} \right) \quad (\text{A.73})$$

$$V_i'' = \frac{r_i^{3/4}}{r_i^{3/4} x_i + r_j^{3/4} x_j} \quad (\text{A.74})$$

A.7 Abraham model

Here, the mathematical formulation of the Abraham model [2] for predicting infinite dilution activity coefficients (IDACs) is presented.

$$\log K_s = c + eE + sS + aA + bB + lL \quad (\text{A.75})$$

$$\ln \gamma_i^\infty = \ln \frac{RT}{P_j^{\text{sat}} V_{m,j} K_s} \quad (\text{A.76})$$

where, the lowercase letters in Eq. A.75 are usually referred to as the descriptors of the solvent, but, in fact, they are fitted to experiments via linear regression [114]. E stands for excess molar refraction computed from refractive index measurements, S is the solute dipolarity-polarizability obtained from gas-liquid chromatography, A and B are the solute hydrogen-bond acidity and basicity, respectively, and L is the solute gas-liquid partition coefficient measured with respect to hexadecane at 298 K. R and T correspond to the gas constant and the temperature, respectively. P_j^{sat} is the saturation pressure of the solvent j and $V_{m,j}$ is the molar volume of j . The c parameter is just another fitting constant.

A.8 Hyperparameters for the isothermal-IDACs GNN model

Some hyperparameters of the proposed isothermal GNN-based model for predicting IDACs were fixed based on experience (Table A.1). The hyperparameters that were tuned using Bayesian optimization are shown in Table A.2. The Bayesian optimization was initialized with 7 samples from a Sobol sequence. Expected improvement was chosen as the acquisition function. The optimization was run for 70 additional iterations. A different random seed was used at each iteration for the random train/validation split using a proportion of 90/10. The hyperparameter optimization was run using `scikit-optimize` [69]. The same node embedding size was used for every message-passing layer. The same training configuration (i.e., Adam optimizer, number of epochs, batch size, learning rate scheduler) was used for each trial in the hyperparameter optimization.

Tab. A.1.: Fixed hyperparameters based on experience.

Hyperparameter	Value
MLP hidden layers	2
Number of epochs	200
Batch size	32

Tab. A.2.: Tuned hyperparameters using Bayesian optimization, exploration bounds and final selected values.

Hyperparameter	Exploration bounds	Selected value
Learning rate	Categorical(0.0001, 0.001, 0.01)	0.001
Dropout probability	Categorical(0.05, 0.1, 0.3, 0.5)	0.1
Message-passing layers	Integer(low=2, high=5)	5
Node embedding size	Integer(low=16, high=64)	30
$\phi_E^{(l)}(\cdot)$ hidden-layer neurons	Integer(low=16, high=64)	64
Neurons first hidden-layer of MLP	Integer(low=32, high=64)	50
Neurons second hidden-layer of MLP	Integer(low=16, high=32)	25

A.9 Distribution of IDAC values in the isothermal 298.15 K data set

In Fig. A.1 the proportion of data points for the train and test sets are shown according to their IDAC value for the data set used in Chapter 3. Additionally, the proportion of IDAC values in the entire data set is shown. Each data set is divided into 100 bins.

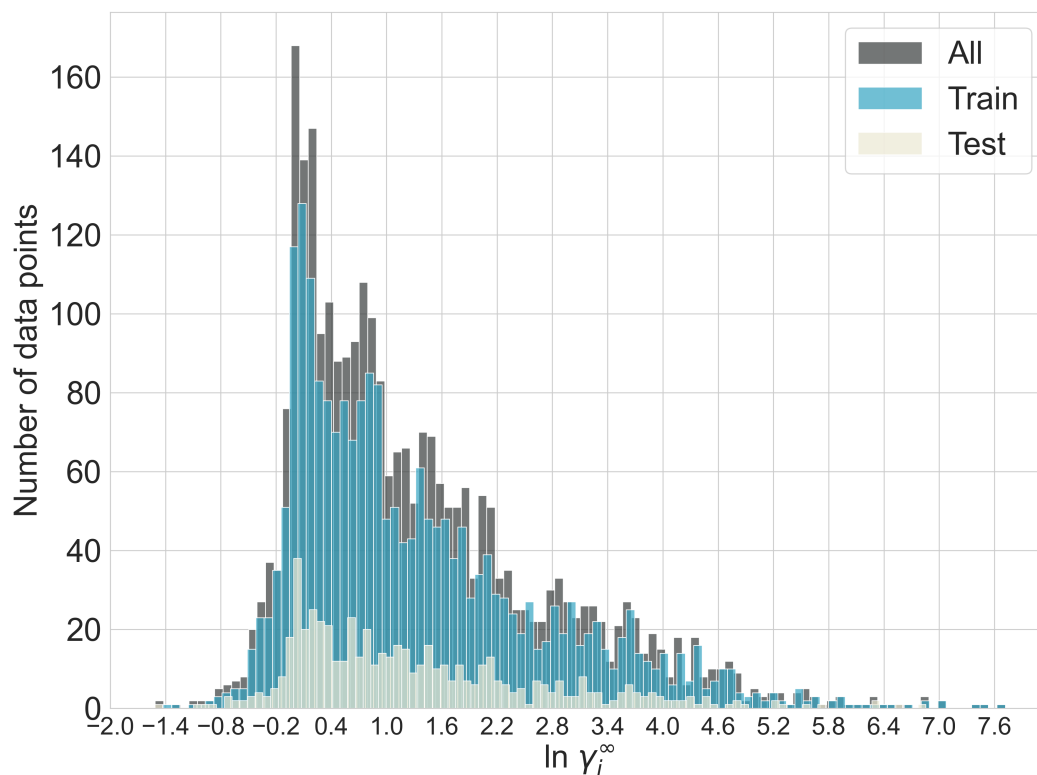


Fig. A.1.: Proportion of data points contained in the train and test data sets, as well as in the entire data set.

A.10 Errors in the Vol. IX of DECHEMA Chemistry Data Series

During the digitization phase of the IDAC collection contained in Vol. IX of DECHEMA Chemistry Data Series [55], some errors were noted and corrected. The following is a description of them containing an internal ID that is used for the database maintenance at the process systems engineering group at MPI-Magdeburg.

- Internal ID: 323 - The formula shown in pages 1599 and 1790 is wrong ($C_7H_{12}O$). The correct formula of the compound according to the DDB databank (DDB No.: 922) is $C_7H_{14}O$. This is also confirmed by PubChem https://pubchem.ncbi.nlm.nih.gov/compound/1_2-Epoxyheptane.
- Internal ID: 475 - The formula shown in pages 1439 and 1805 is wrong ($C_{12}H_{20}O_3$). The correct formula of the compound according to the original paper ([https://doi.org/10.1016/S0021-9673\(01\)81586-7](https://doi.org/10.1016/S0021-9673(01)81586-7)) is $C_{12}H_{20}O_2$ which name corresponds to Linalyl acetate in Table 2 of the mentioned paper.
- Internal ID: 490 - The formula shown in pages 1440 and 1806 is wrong ($C_{16}H_{30}O_2$). The correct formula of the compound according to original paper ([https://doi.org/10.1016/S0021-9673\(01\)81586-7](https://doi.org/10.1016/S0021-9673(01)81586-7)) is $C_{15}H_{28}O_2$ which name corresponds to R_4OCOCH_3 in Table 1 of the mentioned paper.
- Internal ID: 964 - The formula shown in pages 2678 and 2729 is wrong ($C_{12}H_{19}F_6N_3O_4$). The correct formula of the compound according to the DDB databank (DDB No.: 18162) is $C_{12}H_{19}F_6N_3O_4S_2$. This is also confirmed by the name of the compound.
- Internal ID: 1012 - The formula shown in pages 2681 and 2730 is wrong ($C_{14}H_{23}F_6N_3O_4$). The correct formula of the compound according to the name should include "S" in the formula. A search of the compound in the DDB tool (<http://www.ddbst.com/unifacga.html>) also confirms that the formula should be $C_{14}H_{23}F_6N_3O_4S_2$.
- Internal ID: 1182 - The formula shown in page 1816 is wrong ($C_5H_8O_2$). The correct formula of the compound according to the DDB databank (DDB No.: 32277) is $C_5H_6O_2$.
- Internal ID: 1378 - The formula shown in page 1831 is wrong ($C_8H_{14}O$). The correct formula of the compound according to the DDB databank (DDB No.: 22316) is $C_8H_{12}O$.

- Internal ID: 1584 - The formula shown in page 2736 is wrong ($C_7H_9F_6N_3S_2$). The correct formula of the compound according to page 2147 and to the DDB databank is $C_7H_9F_6N_3O_4S_2$.
- Internal ID: 1587 - The formula shown in page 2737 is wrong ($C_7H_{15}N_2O_4S$). The correct formula of the compound according to the name should include "P" in the formula. This is also confirmed in page 2160, showing the correct formula: $C_7H_{15}N_2O_4P$.
- Internal ID: 1601 - The formula shown in pages 2262 and 2738 is wrong ($C_9H_{18}F_6N_2O_4S$). The correct formula of the compound according to the DDB databank (DDB No.: 21225) is $C_9H_{18}F_6N_2O_4S_2$.
- Internal ID: 1622 - The formula shown in pages 2373 and 2739 is wrong ($C_{14}H_{28}BBrN_2O$). The correct formula of the compound according to the DDB tool (<http://www.ddbst.com/unifacga.html>) indicates that the formula should be $C_{14}H_{28}BBrN_2O_2$.
- Internal ID: 1626 - The formula shown in pages 2388 and 2739 is wrong ($C_{16}H_{32}BBrN_2O$). The correct formula of the compound according to the DDB tool (<http://www.ddbst.com/unifacga.html>) indicates that the formula should be $C_{16}H_{32}BBrN_2O_2$.
- Internal ID: 1627 - The formula shown in page 2739 is wrong ($C_{16}H_{32}N_2O_4$). The correct formula of the compound according to the DDB databank (DDB No.: 20036) is $C_{16}H_{32}N_2O_4S$. This is confirmed by the correct formula shown in page 2389.
- Internal ID: 1644 - The formula shown in pages 2462 and 2740 is wrong ($C_{18}H_{36}BBrN_2O$). The correct formula of the compound according to the DDB tool (<http://www.ddbst.com/unifacga.html>) indicates that the formula should be $C_{18}H_{36}BBrN_2O_2$.
- Internal ID: 1674 - The formula shown in pages 2539 and 2741 is wrong ($C_{34}H_{68}F_6NO_4P$). The correct formula of the compound according to the DDB databank (DDB No.: 21227) is $C_{34}H_{68}F_6NO_4PS_2$.

A.11 Details of the DECHEMA data set of IDACs

Fig. A.2 shows the number of data points used in the construction of the DECHEMA data set (i.e., including the repeated measurements of systems at the same conditions) according to the experimental technique. The meaning of the experimental technique identifiers is available in Table A.3.

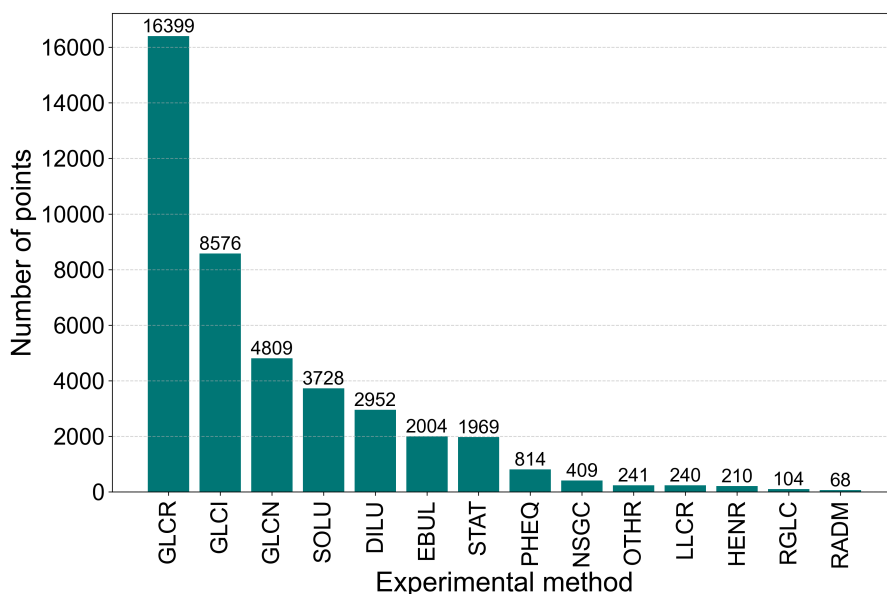


Fig. A.2.: Number of data points used in the construction of the DECHEMA data set according to the experimental technique.

Figures A.3 and A.4 show the distribution of logarithmic IDAC and temperature values in the DECHEMA data set, respectively. The distribution of values is given for the train and test data sets obtained from the stratified splitting.

Fig. A.5 shows the frequency of the 30 most popular chemical classes included in the DECHEMA data set as calculated by the Classyfire [38] ontology. The rest of least frequent classes are combined into the last bar in Fig. A.5 for visualization purposes. However, all chemical classes were considered for the stratified splitting as discussed in Section 3.5.1.

The compounds that were not assigned to a specific chemical class using Classyfire [38] are shown in Table A.4 together with their corresponding “superclass” classification obtained from Classyfire (if available). Most of them correspond to acetylides which are compounds in which one or both hydrogen atoms of ethyne are replaced by a metal or other cationic group.

ID	Meaning
DILU	Dilutor technique
EBUL	Ebulliometry
GLCI	Gas-liquid chromatography with no gas phase correction
GLCN	Gas-liquid chromatography with no specification of gas phase correction
GLCR	Gas-liquid chromatography with gas phase correction
HENR	Derived from Henry coefficients
LLCR	Liquid-liquid chromatography
NSGC	Non-steady-state gas-liquid chromatography
OTHR	Other methods, e.g., isopiestic or dew point technique
PHEQ	Derived from phase equilibrium data at low concentration
RADM	Rayleigh distillation method
RGLC	Relative gas-liquid chromatography
SOLU	Derived from solubility data
STAT	Static method

Tab. A.3.: Meaning of the experimental technique identifiers.

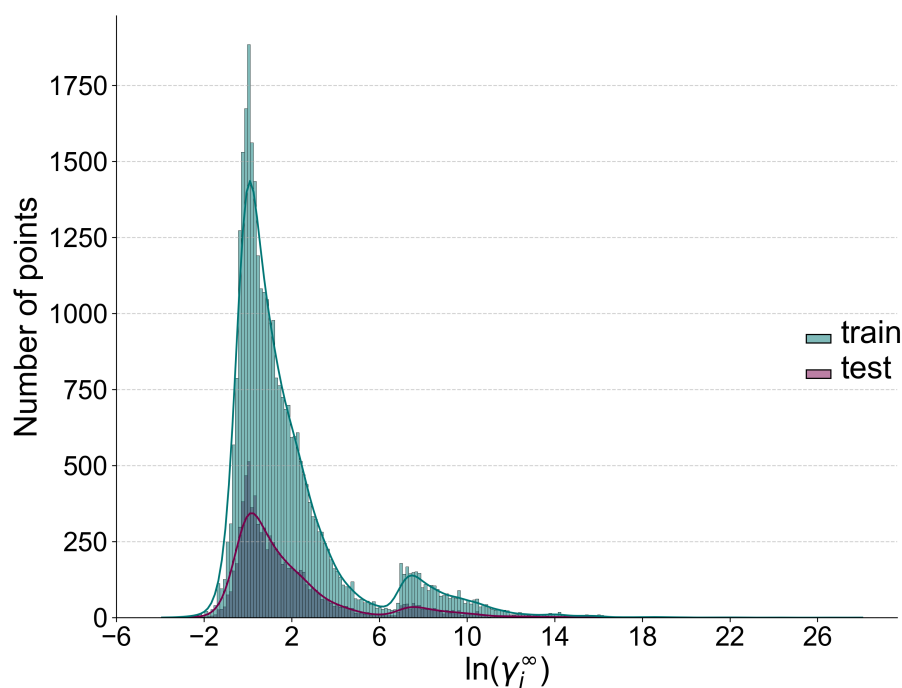


Fig. A.3.: Distribution of logarithmic IDACs in the DECHEMA data set.

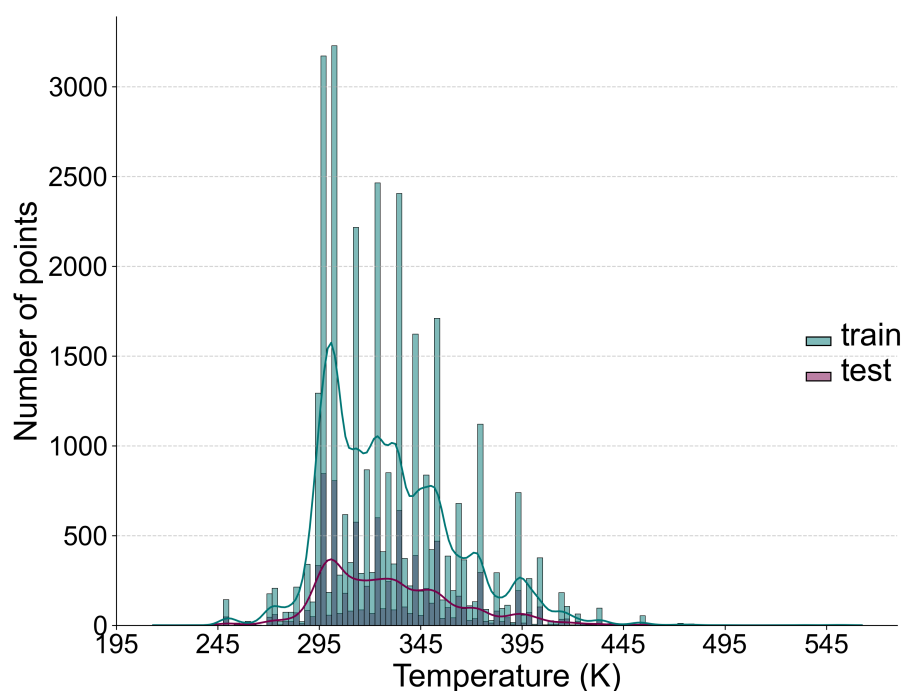


Fig. A.4.: Distribution of temperature values in the DECHEMA data set.

Tab. A.4.: Compounds in the DECHEMA data set that were not assigned to a specific chemical class using Classyfire [38].

#	Compound name	Superclass
1	Propyne	Acetylides
2	1-butyne	Acetylides
3	Germane, tetramethyl	Organometallic compounds
4	Silane, tetramethyl	Hydrocarbon derivatives
5	1-pentyne	Acetylides
6	1-hexyne	Acetylides
7	1-heptyne	Acetylides
8	1,6-heptadiyne	Acetylides
9	1-octyne	Acetylides
10	Silane, tetraethyl	Hydrocarbon derivatives
11	1,8-nonadiyne	Acetylides
12	1-nonyne	Acetylides
13	1-decyne	Acetylides
14	Propanephosphinic acid, dibutyl ester	Organophosphorus compounds
15	Propanoic acid, 3-(2,2,3,3-tetrafluoropropoxy), nitrile	-
16	Propanoic acid, 3-pentoxy, nitrile	-
17	Decanoic acid, 10(9)-perfluorooctyl, methyl ester	-

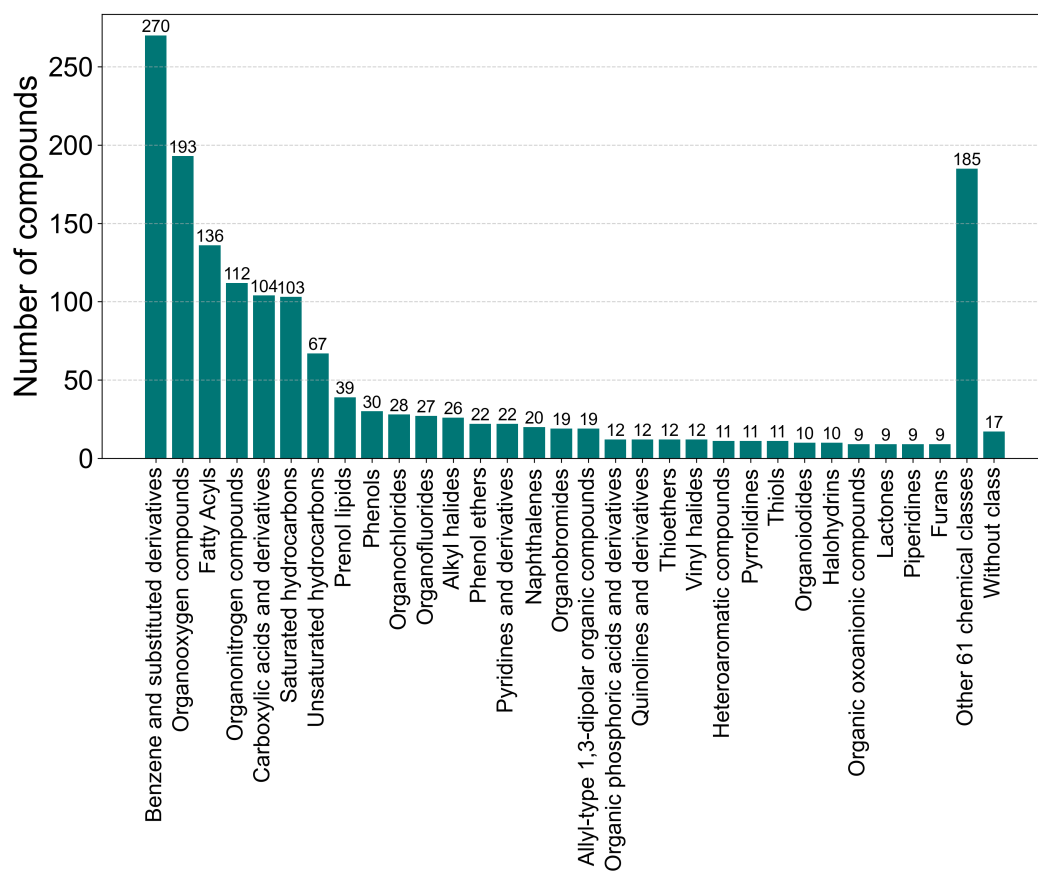


Fig. A.5.: Number of compounds in the DECHEMA data set contained in the 30 most frequent chemical classes computed by Classyfire [38].

A.12 Hyperparameters for e-GNNprev and e-SolvGNN in extended isothermal analysis

The tuning of hyperparameters for e-GNNprev was conducted with `Optuna` [5], specifying 100 trials and utilizing the ranges and scales outlined in Table A.5. A 10-fold cross-validation approach was employed for the tuning process. The chosen hyperparameters for each isothermal study are detailed in Table A.6. Mean Squared Error (MSE) served as the loss function.

Tab. A.5.: Ranges used during the hyperparameter search of e-GNNprev in the isothermal studies.

Hyperparameter	Range	Search scale
Message passing layers	2–5	integer
Dropout ratio	0.05–0.5	uniform
Hidden embedding size	16–256	integer
Learning rate	0.0001–1	loguniform
Units in message passing	8–64	integer
Epochs	100–300	integer
Batch size	4–64	integer
Jumping knowledge	{last, sum, mean}	categorical
Global pooling	{sum, mean, max, set2set}	categorical
Hidden layers final MLP	1–2	integer
Units in final MLP	16–128	integer

Tab. A.6.: Final selected hyperparameters for e-GNNprev in each isothermal study.

Hyperparameter	Final value for T (K)									
	293.15	298.15	303.15	313.15	323.15	333.15	343.15	353.15	373.15	
Message passing layers	4	3	4	2	3	3	4	3	5	
Dropout ratio	0.13	0.05	0.13	0.05	0.05	0.06	0.05	0.07	0.07	
Hidden embedding size	234	41	154	209	72	88	250	64	30	
Learning rate	0.002	0.002	0.011	0.003	0.002	0.011	0.001	0.007	0.009	
Units in message passing	21	31	18	37	12	42	37	39	10	
Epochs	188	239	274	238	168	289	226	144	256	
Batch size	39	59	62	44	53	44	42	53	62	
Jumping knowledge	last	sum	mean	mean	mean	mean	sum	mean	mean	
Global pooling	sum	sum	set2set	sum	sum	sum	sum	sum	mean	
Hidden layers final MLP	2	2	1	2	2	2	2	2	2	
Units in final MLP	105, 74	83, 30	125	110, 112	104, 126	113, 122	105, 18	97, 56	100, 30	

Hyperparameter optimization for e-SolvGNN was performed with `Optuna` [5], employing 100 trials alongside the ranges and scales provided in Table A.7. The optimization process utilized 10-fold cross-validation. The optimized hyperparameters for each isothermal analysis can be found in Table A.8, with Mean Squared Error (MSE) being the chosen loss function.

Tab. A.7.: Ranges used during the hyperparameter search of e-SolvGNN in the isothermal studies.

Hyperparameter	Range	Search scale
Hidden embedding size	16-256	integer
Learning rate	0.0001-1	loguniform
Epochs	100-300	integer
Batch size	4-64	integer

Tab. A.8.: Final selected hyperparameters for e-SolvGNN in each isothermal study.

Hyperparameter	Final value for T (K)								
	293.15	298.15	303.15	313.15	323.15	333.15	343.15	353.15	373.15
Hidden embedding size	242	226	236	186	197	182	252	162	177
Learning rate	0.0004	0.0004	0.0001	0.0002	0.0006	0.0003	0.0006	0.0002	0.0006
Epochs	156	178	287	151	212	299	256	254	260
Batch size	5	12	4	4	9	7	8	4	4

A.13 Hyperparameters for the GH-GNN, GH-SolvGNN, GNNCat and SolvGNNCat models

The hyperparameters for the GNNCat and SolvGNNCat models were optimized using the `Optuna` framework [5]. This optimization process entailed executing 100 trials, with each trial spanning 50 epochs, using hyperparameter ranges as outlined in Tables A.9 and A.10. The selected hyperparameters, after this optimization phase, are also presented in these respective Tables. For the purpose of hyperparameter tuning, 10-fold cross-validation was utilized. Upon determining the final set of hyperparameters, the number of epochs was varied across the values 100, 150, 200, 250 and 300 to find the configuration that most significantly enhanced performance on the validation set. For all models, 250 epochs was found as the best value.

For comparative analysis, the hyperparameters of GH-GNN and GH-SolvGNN were assigned to be equal to those of their counterparts, GNNCat and SolvGNNCat, respectively. To maintain an equivalent number of parameters across the models being compared, the first hidden-layer size of the MLP in GNNCat and SolvGNNCat was adjusted to double the intended hidden embedding size. The total number of trainable parameters amounted to 2,483,580 for both GNNCat and GH-GNN, and 1,798,825 for SolvGNNCat and GH-SolvGNN.

Tab. A.9.: Hyperparameter details for GNNCat.

Hyperparameter	Range	Search scale	Selected value
Hidden embedding size	16-256	integer	113
Learning rate	0.0001-1	log-uniform	0.0002
Batch size	4-64	integer	32

Tab. A.10.: Hyperparameter details for SolvGNNCat.

Hyperparameter	Range	Search scale	Selected value
Hidden embedding size	16-256	integer	193
Learning rate	0.0001-1	log-uniform	0.00012
Batch size	4-64	integer	16

A.14 Errors in the Vol. XIV of DECHEMA Chemistry Data Series

During the digitization phase of the IDAC collection contained in Vol. XIV of DECHEMA Chemistry Data Series [66], some errors were noted and corrected. The following is a description of them containing an internal ID that is used for the database maintenance at the process systems engineering group at MPI-Magdeburg.

- Internal ID: 1 - The chemical formula for glycerol triacetate is presented as $C_9H_{15}O_6$. Nonetheless, the accurate formula ought to be $C_9H_{14}O_6$.
- Internal ID: 2 - The polymer marked as "poly(ϵ -valerolactone)" should actually be termed "poly(δ -valerolactone)," as indicated in the original paper from which the data was extracted [131]. An error in Table 1 of this paper led to the aforementioned typo, suggesting a potential propagation of the mistake from there to the DECHEMA collection.
- Internal ID: 3 - The polymer previously labeled as "polyethylene low-density" has now been annotated as a branched polymer, as indicated by note 86 of the original DECHEMA collection.
- Internal ID: 4 - The polymer listed as "polystyrene, antishock" in the original DECHEMA data collection is identified as a homopolymer. However, this polymer is categorized as a copolymer, commonly referred to as "high impact polystyrene (HIPS)."
- Internal ID: 5 - In the original data collection, the polymers "polyoxyethylene, α,ω -dihydroxy" and "poly(ethylene oxide)" are presented as separate entities. However, considering that one specifies the start and end groups, it is highly probable that both refer to the same chemical structure.

A.15 Derivation of the extended Margules equation

If one considers that the molar excess Gibbs energy g^E of any mixture can be described by a continuous function that is infinitely differentiable, g^E can be expressed as a Taylor series. We will assume here that g^E can be sufficiently well-approximated when truncating the series after the third term. Therefore, for a mixture of N components, its molar excess Gibbs energy can be approximated by

$$g^E(\mathbf{x}) = a_0 + \sum_{i=1}^{N-1} a_i x_i + \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} a_{ij} x_i x_j + \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} \sum_{k=1}^{N-1} a_{ijk} x_i x_j x_k \quad (\text{A.77})$$

where, \mathbf{x} is the vector of molar fractions of $N - 1$ mixture components, and a stands for the corresponding polynomial coefficient according to the subscripts given.

We know from the boundary condition, given in Eq. 2.30, that $g^E = 0$ when $x_N = 1$, and, hence, $a_0 = 0$. Moreover, we know that $(a_i + a_{ii} + a_{iii}) = 0$ when $x_i = 1 \quad \forall i \in \{1, 2, \dots, N - 1\}$. Hence, we can reformulate Eq. A.77 as

$$g^E(\mathbf{x}) = - \sum_{i=1}^{N-1} (a_{ii} + a_{iii}) x_i + \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} a_{ij} x_i x_j + \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} \sum_{k=1}^{N-1} a_{ijk} x_i x_j x_k \quad (\text{A.78})$$

We can then, multiply the first and second summation terms of Eq. A.78 by 1 expressed as the sum of molar fractions to obtain

$$g^E(\mathbf{x}) = - \sum_{i=1}^{N-1} \sum_{j=1}^N \sum_{k=1}^N (a_{ii} + a_{iii}) x_i x_j x_k + \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} \sum_{k=1}^N a_{ij} x_i x_j x_k + \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} \sum_{k=1}^{N-1} a_{ijk} x_i x_j x_k \quad (\text{A.79})$$

To make the expression more concise, all crossed terms in the Taylor polynomial are collected into a single term multiplied by the corresponding (and distinct) constant, so as to have

$$g^E(\mathbf{x}) = - \sum_{i=1}^{N-1} \sum_{j=i}^N \sum_{k=j}^N (a_{ii} + a_{iii}) x_i x_j x_k + \sum_{i=1}^{N-1} \sum_{j=i}^{N-1} \sum_{k=j}^N a_{ij} x_i x_j x_k + \sum_{i=1}^{N-1} \sum_{j=i}^{N-1} \sum_{k=j}^{N-1} a_{ijk} x_i x_j x_k \quad (\text{A.80})$$

This results in a general expression that can be applied to mixtures of N components. The parameters of Eq. A.80 can be related to the binary IDAC values of all mixture component pairs given the relationship

$$RT \ln \gamma_i^\infty = \left(\frac{\partial g^E}{\partial x_i} \right)_{x_j \rightarrow 1, j \neq i} \quad (\text{A.81})$$

Here, we will derive two specific cases:

For **binary mixtures** ($N = 2$) Eq. A.80 reduces to

$$g^E = -a_{111} x_1^2 x_2 + (-a_{11} - a_{111}) x_1 x_2^2 \quad (\text{A.82})$$

we can then apply Eq. A.81 on Eq. A.82 to determine the value of the constants from the IDACs:

$$\left(\frac{\partial g^E}{\partial x_1} \right)_{x_2 \rightarrow 1} = -a_{11} - a_{111} = w_{12} = RT \ln \gamma_{12}^\infty \quad (\text{A.83})$$

$$\left(\frac{\partial g^E}{\partial x_2} \right)_{x_1 \rightarrow 1} = -a_{111} = w_{21} = RT \ln \gamma_{21}^\infty \quad (\text{A.84})$$

Hence, the resulting expression for binary systems is

$$\boxed{g^E = x_1 x_2 (x_1 w_{21} + x_2 w_{12})} \quad (\text{A.85})$$

To obtain an expression for the activity coefficients, we express Eq. A.85 in terms of the excess Gibbs energy $G^E = M g^E$ and then differentiate with respect to the number of moles of the corresponding species:

$$RT \ln \gamma_1 = \frac{\partial G^E}{\partial n_1} = w_{21} \left(\frac{2n_1 n_2}{M^2} - \frac{2n_1^2 n_2}{M^3} \right) + w_{12} \left(\frac{n_2^2}{M^2} - \frac{2n_1 n_2^2}{M^3} \right) \quad (\text{A.86})$$

$$\boxed{RT \ln \gamma_1 = 2w_{21}x_1x_2 + x_2^2w_{12} - 2g^E} \quad (\text{A.87})$$

and similarly for component 2:

$$\boxed{RT \ln \gamma_2 = 2w_{12}x_2x_1 + x_1^2w_{21} - 2g^E} \quad (\text{A.88})$$

Similarly, we can derive the expressions for a **ternary system** ($N = 3$):

$$\boxed{g^E = x_1x_2(x_2w_{12} + x_1w_{21}) + x_1x_3(x_3w_{13} + x_1w_{31}) + x_2x_3(x_3w_{23} + x_2w_{32}) + x_1x_2x_3c_{123}} \quad (\text{A.89})$$

with $c_{ijk} = \frac{1}{2}(w_{ij} + w_{ji} + w_{ik} + w_{ki} + w_{jk} + w_{kj}) - w_{ijk}$. The resulting ternary interaction parameter w_{ijk} is here set to zero as in [6].

$$\boxed{RT \ln \gamma_1 = 2(x_1x_2w_{21} + x_1x_3w_{31}) + x_2^2w_{12} + x_3^2w_{13} + x_2x_3c_{123} - 2g^E} \quad (\text{A.90})$$

$$\boxed{RT \ln \gamma_2 = 2(x_2x_3w_{32} + x_2x_1w_{12}) + x_3^2w_{23} + x_1^2w_{21} + x_3x_1c_{231} - 2g^E} \quad (\text{A.91})$$

$$\boxed{RT \ln \gamma_3 = 2(x_3x_1w_{13} + x_3x_2w_{23}) + x_1^2w_{31} + x_2^2w_{32} + x_1x_2c_{312} - 2g^E} \quad (\text{A.92})$$

Afterword

During the period of development of the work presented in this dissertation, several publications were prepared. The following is a list that describe the publications used as part of the content of this dissertation.

- The methodology presented in Chapter 3 is partially taken from [@5].
- The additional isothermal studies presented in Section 3.5, and the contents of Section 4.1 are based on [@4].
- Section 4.2 and the case-study presented in Section 6.2 are partially based on the work presented in [@2].
- Section 4.3 is based on [@3].
- The methodology presented in Chapter 5 and the case-study presented in Section 6.1 are based on [@6].
- The case-study presented in Section 6.3 is partially taken from [@1].

Publications used for this dissertation

- [@1] Laura König-Mattern, Edgar Ivan Sanchez Medina, Anastasia O. Komarova, Steffen Linke, Liisa Rihko-Struckmann, Jeremy S. Luterbacher, and Kai Sundmacher. “Machine learning-supported solvent design for lignin-first biorefineries and lignin upgrading”. In: *Chemical Engineering Journal* 495 (2024), p. 153524 (cit. on p. 197).
- [@2] Ann-Joelle Minor, Edgar Ivan Sanchez Medina, Ruben Goldhahn, Steffen Linke, Liisa Rihko-Struckmann, and Kai Sundmacher. *Can ionic liquids compete with other solvents in the chemical recycling of Nylon 6?* Under review. 2024 (cit. on p. 197).
- [@3] Edgar Ivan Sanchez Medina, Sreekanth Kunchapu, and Kai Sundmacher. “Gibbs–Helmholtz Graph Neural Network for the prediction of activity coefficients of polymer solutions at infinite dilution”. In: *The Journal of Physical Chemistry A* 127.46 (2023), pp. 9863–9873 (cit. on p. 197).

- [@4] Edgar Ivan Sanchez Medina, Steffen Linke, Martin Stoll, and Kai Sundmacher. “Gibbs-Helmholtz Graph Neural Network: capturing the temperature dependency of activity coefficients at infinite dilution”. In: *Digital Discovery* 2 (3 2023), pp. 781–798 (cit. on p. 197).
- [@5] Edgar Ivan Sanchez Medina, Steffen Linke, Martin Stoll, and Kai Sundmacher. “Graph neural networks for the prediction of infinite dilution activity coefficients”. In: *Digital Discovery* 1 (3 2022), pp. 216–225 (cit. on p. 197).
- [@6] Edgar Ivan Sanchez Medina and Kai Sundmacher. “Solvent pre-selection for extractive distillation using Gibbs-Helmholtz Graph Neural Networks”. In: *Computer Aided Chemical Engineering*. Vol. 52. Elsevier, 2023, pp. 2037–2042 (cit. on p. 197).

Colophon

The writing style of this dissertation was partly refined through the use of two language processing tools: ChatGPT (GPT-3.5, <https://chat.openai.com/>) by OpenAI, and DeepL Write (<https://www.deepl.com/write>). These tools assisted in improving the structure, language, and grammar of the original text. Following this, the author conducted manual proofreading and made additional edits to each paragraph.

This thesis was typeset with \LaTeX 2 ϵ . It uses the *Clean Thesis* style developed by Ricardo Langner. The design of the *Clean Thesis* style is inspired by user guide documents from Apple Inc.

Download the *Clean Thesis* style at <http://cleanthesis.der-ric.de/>.

