國立臺灣大學工學院化學工程學研究所

博士論文

Department of Chemical Engineering

College of Engineering

National Taiwan University

Doctoral Dissertation

開發適用於多成分及複雜化學結構之電腦輔助分子設計方法

Development of Computer-Aided Molecular Design Methods for Multicomponent and Complex Chemical Structures

黃晨軒

Chen-Hsuan Huang

指導教授：林祥泰 博士

Advisor: Shiang-Tai Lin, Ph.D.

中華民國 113 年 7 月

July, 2024

doi:10.6342/NTU202403528

口試委員會審定書

國立臺灣大學博士學位論文
口試委員會審定書
DOCTORAL DISSERTATION ACCEPTANCE CERTIFICATE
NATIONAL TAIWAN UNIVERSITY

開發適用於多成分及複雜化學結構之電腦輔助分子設計方法

Development of Computer-Aided Molecular Design Methods
for Multicomponent and Complex Chemical Structures

本論文係黃晨軒（學號：F07524028）在國立臺灣大學化學工程學研究所完成之博士學位論文，於民國 113 年 7 月 29 日承下列考試委員審查通過及口試及格，特此證明。

The undersigned, appointed by the Department / Graduate Institute of Chemical Engineering on July 29th, 2024, have examined a Doctoral Dissertation entitled above, presented by candidate Chen-Hsuan Huang (Student ID: F07524028), and hereby certify that it is worthy of acceptance.

口試委員 Oral examination committee:

林祥泰　茂英傑　劉德謙

（指導教授 Advisor）

孫珮修　吳台偉　李奕賢

系（所、學位學程）主管 Director: 廖英志

doi:10.6342/NTU202403528

誌謝

我於學士班時曾修習林祥泰老師的物理化學課與計算機程式課。在此後我逐漸對於物理化學理論與電算方法在化學工程上之應用產生興趣，於是我與軒豪同時加入 COMET 實驗室，共同進行電腦輔助分子設計方法的開發。在完成程式雛型的同時，也是我們學士班畢業之時，感謝當時吳台偉教授（Prof. David T. Wu）與謝之真教授擔任我學士論文競賽的口試委員，讓我能順利參賽。畢業後軒豪遠赴美國攻讀博士，而我則留在實驗室攻讀碩博士，持續精進這套技術。

能完成博士論文，我首先需要感謝林祥泰老師的指導以及當初與軒豪一同建立的基礎。軒豪總是充滿效率，讓研究推進更快一些，雖然我們常把自己搞得壓力很大。即使我們和老師都對於這題目相對陌生，老師總是能洞察出題目所衍生出具有研究價值之處，並在我研究碰壁時指引一條明路。老師對於研究課題的思考分析也常讓我覺察到自己和老道的學術研究者差距有多大。老師總是有清楚的大局觀，知道如何區分與彰顯自己的研究題目與其他人有什麼不同。這讓我學到了在埋首於技術細節前應該要求自己也做了這樣的思考。

以學術工作的角度而言，我特別感謝當初帶我新手上路的亞叡、威霖、立行、峻愷、昌哲、旻賢、旻璁。他們給予的幫助包含但不僅限於 Linux 使用、相平衡計算（COSMO-SAC 模型）、量化理論計算、Bash script、電腦機房管理。在學習的過程中，與同儕的討論也讓我在學識上獲益良多。我印象最深刻的討論有幾次，一是與曉丰討論二相熱力學模型（two-phase thermodynamics, 2PT）的理論改進，發現了他在推導新理論時不小心忽略了一些萊布尼茲積分法則（Leibniz integral rule）所衍生的項。二是與力行討論 Maxwell-Stefan 質傳理論所須的物化性如何用 MD/MC 計算出來。三是與德謙研究分子模擬導論課的作業，寫出簡單的 Hartree-Fock 程式。四是和昌哲討論相圖中 miscibility gap 的計算，以及各種化學資訊學軟體的使用細節。五是與肇廷討論深度學習理論與各種模型的工作原理，我特別感謝他分享深厚的見解，為我解惑，同時也覺得他豐富的實作經驗令人讚嘆。六是與亮堯討論 mean-first passage time 與 nucleation 的相關理論，一起理解理論推導中令人困惑的數學式。七是在做中油產學計畫時與李建毅同學（李奕霈老師的學生）討論反應路徑自動探勘的方法。透過閱讀文獻與討論，我覺得自己在那段時間對於化學

doi:10.6342/NTU202403528

反應動力學的理解有了飛越性的提昇。另外，感謝中央大學謝介銘老師與清華大學汪上曉老師在研討會中常駐足聽我報告，並且給予建議。博士生涯的最後一關是口試，在此我也想感謝願意受邀來擔任口委的李奕霈老師、游琇伃老師、吳台偉老師（Prof. David T. Wu）、洪英傑老師以及德謙學長。

在個人雜務與實驗室運作方面，特別感謝前後任助理子芳與琇麗姐幫忙處理許多耗費心神的雜事。特別感謝威霖、立行、昌哲與德謙花費心思維護機房，讓大家能順利產出研究內容。在我接手後更是親身體會管理機房的辛苦之處，接下來大家得依靠亮堯和彥任了。感謝興豪與彥任幫忙分擔環安衛的工作。感謝楊岳、俊杰與晉安幫忙準備口試所需的餐點與飲品。感謝亮堯幫忙紀錄口試問答。感謝力曦日常幫忙跑腿。另外，我也感謝其他前後遇過的實驗室同仁，包括子新、庭嘉、信如、峻承、紹維、奕安、修立、姿妤、藍天、浩恩、柏偉、垚煌、政廷、雲杰、俊霖、晙維、翔云、睿元、郁霖、佳燕、Dalip、Bong-Seop、Aditya、Sanchari。與各位相處是這段漫長時光裡的特別回憶。

最後我想感謝家人支持我讀博士班的決定。當初下決定讀博士班前也是游移很久，念了之後也不乏一些身心上的低潮。在讀博士班高年級的時候，我不知為何地常常抱有一種不耐煩的心態，以及一種研究內容比別人差的感覺。我想有可能是因為這項研究需要冗長的計算時間來產出數據，而產出的數據也不一定展現出明顯的趨勢，讓人可以歸納出知識點。此外，加新功能到既有程式以及重構程式的過程也是 bug 叢生，非常耗費身心與時間。我非常感謝林祥泰老師在我身心遭遇瓶頸時開導我。

doi:10.6342/NTU202403528

中文摘要

本文分為三部分。第一部分闡述電腦輔助分子設計（computer-aided molecular design, CAMD）之概念框架，以及其能如何幫助特用化學品的早期研發。傳統上，特用化學品的研發主要依賴研究人員對問題的經驗，依其化學直覺（chemical intuition）反覆地進行試誤性（trial-and-error）實驗合成與鑑定。由於新課題常與研究員過去經驗有一定差距，在早期研發階段研究方向不明確時，常因冗餘的試驗而造成人力、物力、財力的浪費。電腦輔助分子設計即是想透過電算的方式作為輔助，以改善研發效率。此技術能讓研究者預先得知一小批候選化學物，再鎖定此範圍進行合成與鑑定。在本研究中，我們建立了原子級精細度的電腦輔助分子設計程序。使用者只須給定物化性規格，即可透過最佳化演算法與迭代來設計符合條件的分子。分子設計程序由三要件組成：MARS+分子資料結構（molecular data structure, MDS）、性質預測模型方法、在化學空間（chemical space）搜尋新分子之演算法。

在分子資料結構部分，我們以數學上的圖（graph）來表示一個分子結構。我們預定義了常見原子與一些基團，並指明它們可用的價鍵種類與數目，作為基本元素庫（base element library）。一給定的分子結構轉換成 MARS+資料結構時，其組成原子會被解析為我們預定義的基本元素，並透過八個只包含零與正整數的陣列與兩個字串陣列來描述它們之間的鍵結狀況。其中，元素編號陣列（element indices array）與母元素編號陣列（parent indices array）決定分子內各元素間相對連接關係，鍵級陣列（bond order array）描述上述連接關係之鍵級。元素型別陣列（element type array）記錄各組成原子的種類。元素的異構性（isomerism）則由手性標記陣列（chirality flag array）與兩個順反標記陣列（cis-trans flag array）標示。環號標記陣列（cyclic flag array）與成環鍵結陣列（cyclic bond order array）紀錄分子中之環狀結構。

在性質預測方面，我們基於量化計算軟體，可以算得物質的光電性質，例如 HOMO-LUMO 能隙、絕熱游離能（adiabatic ionization potential）、絕熱電子親和力（adiabatic electron affinity），垂直游離能（vertical ionization potential）、垂直電子親和力（adiabatic electron affinity）、化學硬度（chemical hardness）、親電性指標（electrophilicity index）。此外，也可進行 COSMO 溶合計算，得到分子於溶劑中產

doi:10.6342/NTU202403528

生之屏蔽電荷（screening charge），並輸入至 COSMO-SAC 模型計算活性係數（activity coefficient），應用於相平衡計算。

在搜尋新分子之演算法方面，我們以基因演算法（genetic algorithm, GA）為基底，來對存於 MARS+資料結構中的分子結構做修飾，以產生新分子。其模式主要分為添加（addition）、減去（subtraction）、插入（insertion）、元素改變（element change）、鍵級改變（bond change）、成環（cyclization）、開環（decyclization）、手性反轉（chirality inversion）、順反異構性反轉（cis-trans inversion）、片段交換（crossover）、接合（combination）、成分交換（component switch）。產生的新分子會先進行物化性之計算，並依照適應度函數（fitness function），賦予接近物化性規格要求者較高的適應度（fitness）。最後，以天擇演算法（selection algorithm）決定新分子何者可留存至下一迭代。本研究建立的天擇演算法包含輪轉法（roulette wheel, RW）、模擬退火（simulated annealing, SA）、適應度蒙地卡羅（fitness Monte Carlo, FMC）、非支配排序演算法（non-dominated sorting, NS）。反覆進行「基因演算法-性質預測-天擇演算法」迭代，即可逐步設計出接近物化性規格要求之分子。

本作第二部分以設計新型離子液體作為二氧化碳吸附劑作為範例，展示我們自建的分子設計能因應任務特異性（task-specific）進行設計。在此部份我們使用 COSMO-SAC 模型預測二氧化碳於離子液體的物理吸附溶解度。為了驗證模型的準確度，我們蒐集了 96 種離子液體共 4537 筆實驗數據，並比對其與 COSMO-SAC 模型預測結果的一致性，結果顯示其精確度足夠作為定性或半定量之用。設計出的 3500 種離子液體，有 80 %其碳捕捉的表現與已被文獻報告者相當，而有少數比已知離子液體好許多。分子設計的結果顯示若要將二氧化碳溶解度提高，則離子液體的陰離子基團需要限縮至氟、氯、溴、碘，或者氫氧根離子。

本作第三部分使用 GuacaMol 與 MolOpt 兩套基準套件（benchmark suite）平臺來比較 MARS+與其他生成式模型用於有機分子設計任務時的表現差異。GuacaMol 平臺主要評估效度（effectiveness），亦即足夠長的迭代數下，生成式模型能否達成目標。而 MolOpt 平臺主要評估效率（efficiency），亦即制定非常有限的化學物產生數額度，觀察在額度內所產生的化學物之優選性（optimality）。在 GuacaMol 平臺的比較結果顯示 MARS+的表現位列第二，僅次於 GRAPH_GA 模型。在多數任務中，MARS+和 GRAPH_GA 表現相匹敵，但在搜尋結構異構物

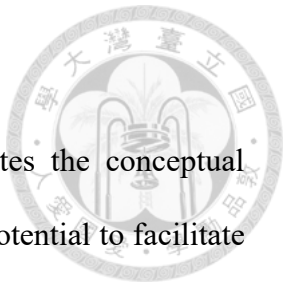
doi:10.6342/NTU202403528

（constitutional isomers）方面明顯比 GRAPH_GA 表現不好。MARS+的片段交換（crossover）操作子經過泛化（generalization）後，可顯著提昇結構異構物的搜尋能力，但同時也會大幅犧牲其在一些單目標任務（single-objective tasks）的表現。在 MolOpt 平臺的比較結果顯示 MARS+的表現位列第三，僅次於第一的 REINVENT 模型與第二的 GRAPH_GA 模型。在多數任務中，MARS+和 GRAPH_GA 表現相匹敵，但在搜尋希樂葆（Celecoxib）藥物分子方面明顯比 GRAPH_GA 表現不好。主因可能在於 GRAPH_GA 有環片段交換（ring crossover）操作子來確保操作前後環的數量未減少。

在展望與未來工作方面主要有四點。第一點是運用 CAMD 於其他化學系統的設計。一些化學系統的設計任務是現行的 MARS+可以做到，或者僅須經由小幅度修改程式即可做到。例如：藥物共晶（pharmaceutical cocrystals）、雙鹽類離子液體（double-salt ionic liquids, DSILs）、深共熔溶劑（deep eutectic solvents, DESs）、光電材料、生物巨分子、高分子聚合物等。第二點是進一步多樣化分子的操作機制在，例如在 MARS+增加環片段交換（ring crossover）操作子。第三點是將分子設計與化工程序設計整合，形成整體的設計方法。第四點是定性比較 MARS+內的各種選擇演算法（selection algorithm），以幫助我們進一步釐清這些演算法的行為。

關鍵字：電腦輔助分子設計、分子表示法、化學物篩選、溶劑、離子液體、碳捕捉、分子生成式模型比較

doi:10.6342/NTU202403528



Abstract

This work is divided into three parts. The first part elucidates the conceptual framework of Computer-Aided Molecular Design (CAMD) and its potential to facilitate the early-stage development of specialty chemicals. Traditionally, the development of specialty chemicals has primarily relied on researchers' experience, involving iterative synthesis and characterization. Given the frequent discrepancies between new challenges and researchers' past experiences, the early development phase often suffers from directionless experimentation, leading to a waste of manpower, materials, and financial resources. CAMD aims to enhance research efficiency by leveraging computational methods to pre-identify a small pool of candidate chemicals for targeted synthesis and characterization. In this study, we have established an atomically detailed CAMD procedure. Users can input the desired physicochemical properties, and the system employs optimization algorithms and iterative processes to design molecules that meet these criteria. The molecular design process comprises three key components: the MARS+ molecular data structure (MDS), property prediction models, and algorithms for searching new molecules in chemical space.

In the molecular data structure component, we represent a molecular structure as a mathematical graph. We predefine common atoms and certain functional groups, specifying their available valence bonds and numbers as a base element library. When a given molecular structure is converted into the MARS+ data structure, its constituent atoms are parsed into our predefined basic elements. Their bonding status is described using eight arrays, containing only zeros and positive integers, along with two string arrays.

For property prediction, we use quantum calculation software to compute the optoelectronic properties of substances, such as the HOMO-LUMO gap, adiabatic

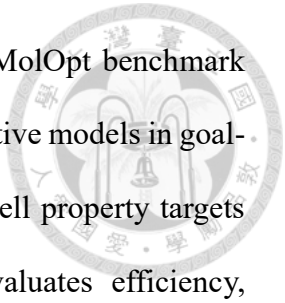
doi:10.6342/NTU202403528

ionization potential, adiabatic electron affinity, vertical ionization potential, vertical electron affinity, chemical hardness, and electrophilicity index. Additionally, COSMO solvation calculations are conducted to obtain the screening charge of molecules in solvents, which is then input into the COSMO-SAC model to calculate activity coefficients, applicable in phase equilibrium calculations.

The algorithm for searching new molecules is based on the Genetic Algorithm (GA), which modifies molecular structures stored in the MARS+ data structure to generate new molecules. Newly generated molecules undergo physicochemical property calculations and are evaluated for fitness based on a fitness function, with those closely matching the desired specifications receiving higher fitness scores. Finally, a selection algorithm determines which new molecules advance to the next iteration. Our selection algorithms include Roulette Wheel (RW), Simulated Annealing (SA), Fitness Monte Carlo (FMC), and Non-dominated Sorting (NS). Repeated iterations of the "Genetic Algorithm - Property Prediction - Selection Algorithm" cycle progressively yield molecules that closely meet the specified physicochemical criteria.

The second part of this work demonstrates the application of our molecular design framework to develop novel ionic liquids as CO_2 adsorbents. In this section, we use the COSMO-SAC model to predict the physical absorption solubility of CO_2 in ionic liquids. To validate the model's accuracy, we collected 4537 experimental data points for 96 ionic liquids and compared them with the COSMO-SAC model predictions. The results show sufficient accuracy for qualitative or semi-quantitative purposes. Among the 3500 designed ionic liquids, 80% exhibited CO_2 capture performance comparable to those reported in the literature, with a few significantly outperforming known ionic liquids. The design results suggest that enhancing CO_2 solubility requires constraining the anionic groups of the ionic liquids to fluoride, chloride, bromide, iodide, or hydroxide ions.

doi:10.6342/NTU202403528



In the third part of this study, we utilized the GuacaMol and MolOpt benchmark suites to assess the performance of MARS+ compared to other generative models in goal-directed tasks. GuacaMol evaluates effectiveness, measuring how well property targets are achieved over a sufficient number of iterations. MolOpt evaluates efficiency, assessing the optimality of generated species within a limited number of iterations. In GuacaMol, MARS+ ranked 2nd, closely behind the GRAPH_GA model. In MolOpt, MARS+ ranked 3rd, following the REINVENT model (1st) and GRAPH_GA (2nd). Generalizing the crossover operator in MARS+ significantly enhances its capability to search for constitutional isomers, albeit at the cost of performance in single-objective tasks. The ring crossover operator in GRAPH_GA appears to be a significant factor contributing to performance differences between MARS+ and GRAPH_GA.

There are four potential avenues for future research. First, extending CAMD applications to other chemical systems where current MARS+ capabilities suffice or require minor program modifications, such as pharmaceutical cocrystals, double-salt ionic liquids (DSILs), deep eutectic solvents (DESs), optoelectronic materials, biomolecules, and polymers. Second, further diversifying molecular operational mechanisms, including integrating a ring crossover operator into MARS+. Third, integrating molecular design with chemical process design to make the design tasks more realistic. Fourth, conducting qualitative comparisons of various selection algorithms within MARS+ to gain deeper insights into their behaviors.

Keywords: Computer-aided molecular design, molecular representation, chemical screening, solvent, ionic liquid, carbon capture, comparisons among molecular generative models.

doi:10.6342/NTU202403528

Table of Contents

口試委員會審定書 i

誌謝 ii

中文摘要 iv

Abstract vii

Table of Contents x

List of Figures xv

List of Supplementary Figures xxi

List of Tables xxiii

List of Supplementary Tables xxiv

Chapter 1. Introduction 1

1.1. Chemical Products and Innovations 1

1.2. Molecular Databases and Chemical Space 4

1.3. Computer-Aided Molecular Design (CAMD) 5

1.3.1. Bidirectional Relation: Molecular Structure and Properties 6

1.3.2. Mathematical Formulation of a CAMD Problem 8

1.4. The Purposes of This Work and an Outline 10

Chapter 2. Generic Theory 15

2.1. Mixed-Integer Non-Linear Programming (MINLP) 15

2.2. Mathematical Methods for Solving MINLP Problems 17

2.2.1. Continuously Differentiable Problem with Integer Variables 18

2.2.2. Nondifferentiable Problem with Complicated Discrete Variables 21

2.3. Components of Computer-Aided Molecular Design (CAMD) 24

doi:10.6342/NTU202403528

2.3.1. Molecular Data Structure: Chemical Representations 25
2.3.2. Forward Algorithm: Property Predictions Methods 26
2.3.3. Reverse Algorithm, Part (I): Generative Algorithms 31
2.3.4. Reverse Algorithm, Part (II): Selection Algorithms 34
Chapter 3. Constructing a Program for Conventional CAMD 37
3.1. Chemical Representation: MARS+ Package 37
3.1.1. The Library of Base Elements 41
3.1.2. Molecular Data Structure (MDS) 43
3.2. Forward Algorithm: Property Prediction Models 50
3.2.1. COSMO-SAC Activity Coefficient Model 52
3.2.2. Electronic Properties from Quantum Simulations 61
3.2.3. Synthetic Accessibility Score (SAscore) 66
3.2.4. Synthetic Complexity Score (SCscore) 68
3.3. Reverse Algorithm (I): MARS+ Package 71
3.3.1. Structure Manipulations – uni-molecular operations 71
3.3.2. Structure Manipulations – bi-molecular operations 75
3.3.3. Structure Manipulations – bi-supermolecular operations 77
3.3.4. Transformation of SMILES into MDS (smi2mds) 79
3.3.5. Transformation of MDS into SMILES (mds2smi()) 80
3.4. Reverse Algorithm (II): Selection Algorithms 81
3.4.1. Fitness Function 81
3.4.2. Selection Algorithms 83
Chapter 4. Intrinsic Performance of MARS+ based CAMD 89
4.1. Exhaustive Structure Operations on Every Possible Point 89
4.1.1. Insertion 90

doi:10.6342/NTU202403528

4.1.2. Cyclization 92
4.1.3. Decyclization 94
4.1.4. Cis-trans inversion 94
4.1.5. Chirality inversion 95
4.1.6. Crossover 96
4.1.7. Combination 97
4.1.8. Component Swap 98
4.2. Chemical Space Exploration via Iterative Enumeration 99
4.3. Can MARS+ Produce Well-known Chemicals? 103
Chapter 5. Design of Novel ILs for CO_2 Capture 106
5.1. A Review of Theoretical and Application Insights 106
5.2. Thermodynamic Modeling 110
5.3. Validation of COSMO-SAC Predictions 115
5.4. IL Screening Using Experimentally Validated Ions 121
5.5. Computational Details of IL Design Using CAMD 131
5.6. CAMD Results 135
Chapter 6. Rule-based vs. AI-based CAMD 147
6.1. AI-based Generative Models for CAMD 147
6.2. Benchmarks for Comparing Rule-based and AI-based CAMD 151
6.3. GuacaMol: Effectiveness of MARS+ and Other Baseline Models 154
6.4. MolOpt: Efficiency of MARS+ and Other Baseline Models 161
Chapter 7. Conclusions 168
Chapter 8. Prospects and Future Work 170
8.1. Applications to Other Chemical Mixture Systems 170
8.2. Enriching the Mechanisms for Molecular Manipulations 171

doi:10.6342/NTU202403528

8.3. Integrated Computational Molecular-Process Design 173
8.4. Qualitative Comparisons for Implemented Selection Algorithms 174
Appendix A. Supplementary Tables to the Main Texts 175
Appendix B. Supplementary Figures to the Main Texts 195
Appendix C. Optimality in Non-linear Programming 208
C.1. Optimality Conditions in Unconstrained Optimizations[393] 208
C.1.1. First-order Necessary Conditions 208
C.1.2. Second-order Necessary Conditions 208
C.1.3. Second-order Sufficient Conditions 208
C.1.4. A Short Proof 209
C.2. Optimality Conditions in Constrained Optimizations[37, 394] 209
C.2.1. First-order Necessary Condition (Karush-Kuhn-Tucker, KKT) 210
C.2.2. First-order Sufficient Condition (Karush-Kuhn-Tucker, KKT) 213
C.2.3. Second-order Necessary Condition 214
C.2.4. Second-order Sufficient Condition 214
C.3. Tangent Cone and Feasible Directions[393] 215
C.4. Gordan's Theorem[394] 217
C.5. Lagrangian Duality Problem[394] 217
C.6. Nonlinear Duality Theorem[394] 218
Appendix D. Solving Non-linear Programming Problems[68] 219
D.1. Sequential Quadratic Programming (SQP) Method 219
Appendix E. Generalized Benders Decomposition[37, 69, 70] 223
E.1. Problem Projection 223
E.2. Dual Problems of the Projected Problems 224
E.3. Formulation of GBD Form 224

doi:10.6342/NTU202403528

E.4. GBD Algorithm 226

Appendix F. Theories of Some AI-based Generative Models 228

F.1. Neuron and Neural Network (NN) 228

F.2. RNN-based Chemical Semantic Model 230

F.3. VAE-based Latent Variable Model 238

F.4. Transformer Architecture with Self-Attention Mechanism 244

References 247

doi:10.6342/NTU202403528

List of Figures

Figure 1.3-1. The bidirectional relationships between molecular structure and molecular properties: forward algorithm vs. reverse algorithm. 7

Figure 2.2-1. The branch-and-bound process for solving CDMINLP-P1. 20

Figure 2.2-2. The flowchart of solving MINLP problem by GBD method. 23

Figure 2.3-1. The four components of rule-based computer-aided molecular design. ...24

Figure 2.3-2. Each methodology for property prediction is suitable for describing physicochemical phenomena under specific scales of time and length. Reprinted with permission from the reference[86]. Copyright 2009 Elsevier Ltd. 27

Figure 2.3-1. The four components in rule-based CAMD framework. 37

Figure 3.1-1. The architecture of MARS+ package. Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.....40

Figure 3.1-2. The attribute settings for the base element *1,3-dimethylimidazolium* (*id*=36). The last carbon in suffix uses a double bond to connect with [N+], forming a ring with 99999 as its default ring number. Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society. 43

Figure 3.3-1. Illustration of the nine uni-molecular operations. Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society. 74

Figure 3.3-2. Illustration of the two bi-molecular operations. The crossover point is represented by the scissor symbol, while the combination point is denoted by the brown arrow. Adapted with permission from the reference[163]. Copyright 2023 American Chemical Society. 76

Figure 3.3-3. Illustration of the component swap operation. Reprinted with permission

doi:10.6342/NTU202403528

from the reference[163]. Copyright 2023 American Chemical Society. 78

Figure 3.4-1. The relationship between fitness (eq. (3.4–1)) and the mean deviation of properties from targets $\Delta j\mathrm{mi}, \mathrm{si}; t$ with parameter A=6, B=5, C=4, and D=3. 83

Figure 3.4-2. Schematic diagram for Pareto frontier. The number of properties is reduced to two ($m = 2$) for illustration. 87

Figure 4.1-1. Exemplary ionic liquids (a) and (b) used for demonstrating the test of exhaustive single operation. Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society. 89

Figure 4.1-2. The 31 result cations produced from *insertion* operation on the double bonds between the 4th and the 5th element of *cation (a)*. (U) denotes a unique species among the cations shown here. (Caption: element index of element I, element index of element II, ID of the introduced element, bond order of the introduced element with the parent element, bond order of the introduced element with the descendant element). Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.....91

Figure 4.1-3. The 13 result cations produced from *cyclization* operation on *cation (a)*. (U) means a unique species among the cations shown here. (Caption: element index of element I, element index of element II, cyclic bond order in between). Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society. 93

Figure 4.1-4. The structure of *cation (a)* before and after the destruction of C3-N6 ring bond (cyclic flag = 1). Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society. 94

Figure 4.1-5. The structure of *cation (a)* before and after the inversion of cis-trans

doi:10.6342/NTU202403528

isomerism of the 9th element (Caption: element index of the subject element, flag type of the subject element, flag before inversion, flag after inversion) Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society. 95

Figure 4.1-6. The structure of *cation (a)* before and after the inversion of cis-trans isomerism of the 7th element. (Caption: element index of chiral center, chirality flag before operation, chirality flag after operation) Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.....96

Figure 4.1-7. The 4 pairs of cations generated from applying *crossover* operation on the *cation (a)* and *cation (b)*, with crossover point of *cation (a)* fixed at the double bond between its 4th and 5th element. (Caption: crossover point for *cation (a)*, crossover point for *cation (b)*) Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.....97

Figure 4.1-8. The 18 cations generated from applying *combination* operation on *cation (a)* and *cation (b)*, with the 3rd element of *cation (a)* picked as the combination point. (U) means a unique species among the cations shown here. (Caption: element index of the combination point in *cation (a)*, element index of the combination point in *cation (b)*, bond order in between). Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.....98

Figure 4.1-9. The *IL (a)* and *IL (b)* after *component swap* operation. Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.....99

Figure 4.2-1. The number of successful operations and newly generated unique species

doi:10.6342/NTU202403528

per iteration. Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.....101

Figure 4.2-2. The number of successful operations, factorized into the contribution from each operation and each iteration. Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.....102

Figure 4.2-3. The number of novel unique molecules, factorized into the contribution from each operation and each iteration. Every species is credited to the operation responsible for its first appearance. Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.....103

Figure 4.3-1. The intermediate products in the total synthesis scheme of Oseltamivir proposed by E.J. Corey et al.[197, 198] The green numbers are element indices. Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society. 105

Figure 5.3-1. Comparison of COSMO-SAC predicted CO_2 solubility in ionic liquids (ILs) with VLE experimental data. 119

Figure 5.3-2. Comparison of COSMO-SAC predicted Henry's constant of CO_2 in ionic liquids (ILs) with experimental data. 120

Figure 5.3-3. Comparison of COSMO-SAC predicted Henry's constant of CO_2 in ionic liquids (ILs) with experimental data. 121

Figure 5.4-1. Solubility of CO_2 in screened ILs. A dot in a cell indicates the presence of at least one VLE experimental data point for the CO_2-IL system. 123

Figure 5.4-2. Reciprocal SAscore of screened ILs. Red cells indicate high synthetic accessibility (or low structural complexity). A dot in a cell indicates the presence of at least one VLE experimental data point for the CO_2-IL system. 124

doi:10.6342/NTU202403528

Figure 5.4-3. Reciprocal SCscore of screened ILs. Red cells indicate low synthetic complexity. A dot in a cell indicates the presence of at least one VLE experimental data point for the CO_2-IL system 125

Figure 5.4-4. Absorption free energy ($\Delta Gi/Sabs$) of CO_2 in the screened ILs. A dot in a cell indicates the presence of at least one VLE experimental data point for the CO_2-IL system. 127

Figure 5.4-5. Absorption enthalpy ($\Delta Hi/Sabs$) of CO_2 in the screened ILs. A dot in a cell indicates the presence of at least one VLE experimental data point for the CO_2-IL system. 128

Figure 5.4-6. Absorption enthalpy ($\Delta Si/Sabs$) of CO_2 in the screened ILs. A dot in a cell indicates the presence of at least one VLE experimental data point for the CO_2-IL system. 129

Figure 5.4-7. Reciprocal absorption-desorption index (ADI) of CO_2 in the screened ILs. The darker red indicates the desirable performance. A dot in a cell indicates the presence of at least one VLE experimental data point for the CO_2-IL system. 131

Figure 5.5-1. Flow diagram of the CAMD algorithm developed in this work. 134

Figure 5.6-1. Evolution trajectory of population in terms of mean, quartiles, maximum, and minimum value of CO_2 solubility for (a) task 1 and (b) task 2. 137

Figure 5.6-2. Evolution trajectory of the number of cation, anion, and IL species existing in population for (a) task 1 and (b) task 2.....138

Figure 5.6-3. Comparison between H_i-derived and VLE-based CO_2 solubility in each of the designed IL species in (a) task 1 and (b) task 2. The blue dots are the ILs specified in the initial population, and the orange dots are the designed ILs. 140

doi:10.6342/NTU202403528

Figure 5.6-4. The distribution of accumulated new ILs species per 100 generations against the CO_2 solubility provided by the ILs in (a) task 1 and (b) task 2. The inset figure shows the regime of higher CO_2 solubility. 142

Figure 6.1-1. The flowchart for computer-aided molecular design based on SMILES representation and early architecture of RNN[277]. 149

Figure 6.1-2. The flowchart for computer-aided molecular design based on early architecture of VAE[281]. 150

Figure 6.3-1. Performance of 5 baseline models in GuacaMol benchmark, MARS+, and MARS+_modcross. 160

Figure 6.4-1. The steps of (a-c) plain crossover and (d-f) ring crossover in GRAPH_GA. Ring crossover requires two cuts: one at a specified bond and another at adjacent bonds or bonds separated by one bond. In step (f), the resulting ring fragments from step (e) are paired and connected using ring bonds to ensure the number of rings in each molecule remains unchanged after operation. This figure is reproduced from reference[110] with permission from the Royal Society of Chemistry..... 166

Figure 6.4-2. Performance of 25 baseline models in MolOpt benchmark and MARS+_modcross. 167

doi:10.6342/NTU202403528

List of Supplementary Figures

Figure B1. Constructing a programmatic sequence of molecular operations to mimic the Oseltamivir synthesis pathway of E.J. Corey et al. The caption under a structure indicates the operation to bring the previous structure to current one. Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society. 195

Figure B2. Constructing a programmatic sequence of molecular operations to mimic the Oseltamivir synthesis pathway of E.J. Corey et al.: the variation of SCscore and SAscore with respect to reaction steps. Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society. 196

Figure B3. Reciprocal absorption-selectivity-desorption index (ASDI, see eq (5.2–11)) of CO_2 over N2 in the screened ILs. The darker red indicates the desirable performance. A dot in a cell indicates the presence of at least one VLE experimental data point for the CO_2-IL system. 197

Figure B4. Reciprocal absorption-selectivity-desorption index (ASDI, see eq (5.2–11)) of CO_2 over CH4 in the screened ILs. The darker red indicates the desirable performance. A dot in a cell indicates the presence of at least one VLE experimental data point for the CO_2-IL system. 198

Figure B5. Reciprocal absorption-selectivity-desorption index (ASDI, see eq (5.2–11)) of CO_2 over CO in the screened ILs. The darker red indicates the desirable performance. A dot in a cell indicates the presence of at least one VLE experimental data point for the CO_2-IL system. 199

Figure B6. Reciprocal absorption-selectivity-desorption index (ASDI, see eq (5.2–11)) of CO_2 over H_2O in the screened ILs. The darker red indicates the desirable

doi:10.6342/NTU202403528

performance. A dot in a cell indicates the presence of at least one VLE experimental data point for the CO_2-IL system. 200

Figure B7. Reciprocal absorption-selectivity-desorption index (ASDI, see eq (5.2–11)) of CO_2 over O_2 in the screened ILs. The darker red indicates the desirable performance. A dot in a cell indicates the presence of at least one VLE experimental data point for the CO_2-IL system. 201

Figure B8. Selectivity of CO_2 over CH_4 (eq (5.2–8)) in the screened ILs. The darker red indicates the desirable performance. A dot in a cell indicates the presence of at least one VLE experimental data point for the CO_2-IL system. 202

Figure B9. Selectivity of CO_2 over CO (eq (5.2–8)) in the screened ILs. The darker red indicates the desirable performance. A dot in a cell indicates the presence of at least one VLE experimental data point for the CO_2-IL system. 203

Figure B10. Selectivity of CO_2 over H_2 (eq (5.2–8)) in the screened ILs. The darker red indicates the desirable performance. A dot in a cell indicates the presence of at least one VLE experimental data point for the CO_2-IL system. 204

Figure B11. Selectivity of CO_2 over H_2O (eq (5.2–8)) in the screened ILs. The darker red indicates the desirable performance. A dot in a cell indicates the presence of at least one VLE experimental data point for the CO_2-IL system. 205

Figure B12. Selectivity of CO_2 over N_2 (eq (5.2–8)) in the screened ILs. The darker red indicates the desirable performance. A dot in a cell indicates the presence of at least one VLE experimental data point for the CO_2-IL system. 206

Figure B13. Selectivity of CO_2 over O_2 (eq (5.2–8)) in the screened ILs. The darker red indicates the desirable performance. A dot in a cell indicates the presence of at least one VLE experimental data point for the CO_2-IL system. 207

doi:10.6342/NTU202403528

List of Tables

Table 2.3-1. Exemplary representations for methanol molecule. 26

Table 2.3-2. A summary for the aforementioned methods for property predictions. 30

Table 2.3-3. The three types of the generative algorithms. 33

Table 3.1-1. The attributes of a base element. 41

Table 3.1-2. The molecular data structure (MDS) in MARS+ package 48

Table 3.2-1. The property estimation method incorporated in this work 50

Table 3.2-2. The RDKit descriptors[171] incorporated in this work. 51

Table 3.2-3. The dimensions of every layer in SCscore model. 69

Table 3.4-1. Calculation of crowded distance for chemical $\boldsymbol{u1}$. 88

Table 5.1-1. Typical subject gas and operating condition for carbon capture. 107

Table 5.3-1. The accuracy of COSMO-SAC prediction: Henry's constant of CO_2 in ILs. 117

Table 5.3-2. The accuracy of COSMO-SAC prediction: CO_2 solubility in ILs. 118

Table 5.4-1. Some potentially promising ILs discovered from screening method. 122

Table 5.5-1. Summary of the settings for the two CAMD tasks. 135

Table 5.6-1. The optimal species ($xCO2VLE \geq 0.07$) of task 1 143

Table 5.6-2. The optimal species ($xCO2VLE \geq 0.08$) of task 2 144

Table 6.2-1. The significance of the distribution-learning metrics. 152

Table 6.2-2. A survey of benchmark suites and the metrics provided. 154

Table 6.3-1. Goal-directed tasks in GuacaMol.[111] 156

Table 6.3-2. The five baseline models in GuacaMol.[111] 158

Table 6.4-1. The 25 baseline models in MolOpt.[317] 162

doi:10.6342/NTU202403528

List of Supplementary Tables

Table A1. A survey of molecular and property databases. Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.....175

Table A2. A survey of chemical space exploration and size estimation research.....176

Table A3. A survey of work applying CAMD for chemical engineering problems*. Reprinted with permission from the reference[264]. Copyright 2018 American Chemical Society. 177

Table A4. The attributes of neutral base elements. Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.....184

Table A5. The attributes of cation base elements. Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.....186

Table A6. The attributes of anion base elements. Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.....187

doi:10.6342/NTU202403528

Chapter 1. Introduction

1.1. Chemical Products and Innovations

The chemical products can be roughly classified into three groups: fine chemicals, specialty chemicals, and commodities.[1] Fine chemicals are high-purity substances manufactured in small quantities (approximately 1000 tons per year) and sold at premium prices (exceeding $10 per kilogram). High-purity electronic chemicals are examples for fine chemicals. In the semiconductor industry, hydrofluoric acid is widely used for cleaning or etching the wafer. If its purity does not meet the standard, the semiconductor product would suffer from contamination issues and yield losses.

Commodities are produced in large volumes using highly standardized processes and sold at low prices (< $1/kg). Plastics, petrochemicals, fibers, monomers, and other basic chemicals (e.g. methanol, acetic acid, and sulfuric acid, etc.) are typical commodities. The wide range of commodities covers the needs from chemical manufacturers to end consumers. For instance, petrochemicals and monomers are important ingredients for midstream and downstream chemical manufacturers. Many plastic products are sold to end consumers, as we use them widely in our daily life.

Specialty chemicals are typically mixtures of various commodities and fine chemicals, distinguished by their performance properties in specific applications. A formulated drug, with active pharmaceutical ingredients (APIs) derived from fine chemicals, serves as an example of a specialty chemical. In fact, it may not be feasible to draw a clear dividing line between commodities and specialty chemicals. It is suggested that a volume of 1000 tons/year and a price of $10/kg can be used as criteria, although this is somewhat arbitrary.[1] On the other hand, whether a chemical is recognized by its

doi:10.6342/NTU202403528

performance properties may not provide a clear dividing line between fine chemicals and specialty chemicals. In fact, a fine chemical might also be used for particular purposes if its performance properties meet the requirements. Especially, the performance properties of pure substances are often useful benchmarks in preliminary research and development (R&D) studies.

When developing new organic compounds for use as semiconductor materials or optoelectronic materials, exciton binding energy may be the key property for determining their uses. Compounds with low exciton binding energy are relatively suitable for photovoltaic cell applications, while those with a larger exciton binding energy are suitable for light-emitting applications.[2]

In the cracking process of the petrochemical industry, various products can form an azeotropic mixture under certain compositions. Since the vapor composition and the liquid composition are identical for an azeotrope at thermal equilibrium, simple distillation can no longer separate its components. A proper entrainer, which can be either a pure substance or a mixture, can be a solution to this problem. Adding it to the azeotrope can change the activity coefficients of components, leading to a variation of their relative volatility. As a result, several light components can be separated from the azeotrope in advance.[3]

When designing the chemical formulation for a battery, the choice of proper additives for electrolytes can improve performance or safety. Many of the known additives are organic compounds (e.g. vinylene and maleimide derivatives), and they work based on their redox potential. For instance, some additives are able to form a film on the electrode after being reduced, which minimizes the resistance for charge transfer. A spontaneous reaction is anticipated when the redox potential of the additive exceeds that of the electrolyte. Additionally, an additive with high electrochemical reversibility

doi:10.6342/NTU202403528

and a redox potential slightly higher than the cathode's maximum operating potential can offer overcharge protection.[4]

Although both of specialty and fine chemicals are high-value, their market sizes are out of proportion. Fine chemicals only take up around 4% of the global chemical market, while the specialty chemical take up around 55%.[1] Specialty chemicals provide much more value-added opportunities than fine chemicals in terms of the concept of chemical space (elaborated in section 1.2). The full chemical space refers to the set that contains all the theoretically possible chemicals. Fine chemicals are represented by a subset mostly composed of pure substances, whereas specialty chemicals are represented by a subset mostly composed of mixtures. The specialty chemicals have a wider coverage of the chemical space than fine chemicals due to the variety of chemical compositions. It turns out that specialty chemicals are often the focus of chemical innovations.

In recent years, the development of novel specialty chemicals, functional materials, and drugs has gained increasing importance for the chemical industry, as evidenced by the growing global revenue in these sectors.[5] Aside from commercial considerations, these innovations hold significant promise as solutions to some of humanity's most pressing challenges in the 21st century. These include the discovery of medicines to combat pandemics[6], the development of efficient energy storage materials[7], and advancements in carbon capture and storage technologies[8]. Historically, the research and development (R&D) of novel specialty chemicals has heavily relied on researchers' expertise and existing chemical databases. To identify potential chemical candidates with desired performance characteristics, researchers traditionally employed a trial-and-error approach based on experience and chemical intuition. While conceptually straightforward, this methodology suffers from significant drawbacks. It is inherently time-consuming and labor-intensive, with efficiency often hampered by unclear strategic direction, budgetary

doi:10.6342/NTU202403528

constraints, and limitations in domain-specific knowledge. The lengthy commercialization process for new specialty chemicals further underscores the inefficiency of this traditional approach. Literature suggests timelines ranging from 6 to 20 years[9, 10], with product and technology development itself consuming 4 to 15 years[9, 10].

1.2. Molecular Databases and Chemical Space

Molecular structure and property databases (summarized in **Table A1**) play a crucial role in screening chemical candidates and constructing correlative models to understand structure-property relationships. They can serve as a knowledge source complementing the researcher's experience. Based on the molecular properties provided in the databases, one may pre-screen a set of chemicals and use them as chemical candidates or as the precursors for formulating novel mixtures. However, access to some large molecular databases may require a license fee. For instance, LOLI database[11] and Beilstein database[12] fall under this category. LOLI database provides the regulatory data (e.g. toxicological and pharmacological data) of more than 600 thousand species. Beilstein database contains 9.8 million substances and 10.3 million chemical reactions. On the other hand, ChemSpider[13] and PubChem[14] are free of charge. ChemSpider covers 123 million chemical structures, and PubChem contains 115 million validated compounds. PubChem database also provides thermal chemical properties (e.g. vapor pressure, Henry's law constant, heat of vaporization) for many chemical species.

In general, most of the large molecular databases, such as those named above, offer relatively complete information of experimental density, boiling point, melting point. As for other properties, one might need to consult a specialized database. For example, the GDB database[15-18], covering around 167 billion organic molecules, is specialized for

doi:10.6342/NTU202403528

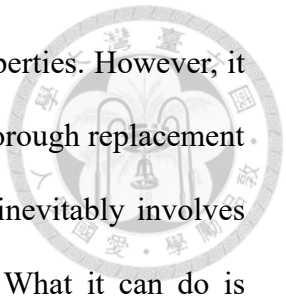
providing ab-initio calculation results of electronic properties, such as the lowest unoccupied orbital (LUMO), the highest occupied orbital (HOMO), dipole moment. The Dortmund databank[19, 20] provides many types of thermodynamic experimental data, including vapor-liquid equilibrium (VLE) data for about 44,000 mixtures and the liquid-liquid equilibrium (LLE) data for about 41,000 mixtures. However, it is essential to recognize that these existing databases likely represent only a small fraction of the vast *chemical space* and the diverse range of molecular properties.[21-24]

In the broadest context, *chemical space* refers to the collection of all theoretically possible chemicals, including thermodynamic mixtures and composite materials. Nevertheless, researchers in a particular scientific domain often narrow down the chemical space to a subset they are interested in, e.g. organic chemical space and drug-like chemical space.[25] Regarding to these two subsets, several studies[26-29] have provided estimations of their size under different additional constraints (e.g. number of heavy atoms, types of constituent atoms, and molecular weight, etc.), as summarized in **Table A2**. In particular, the size of organic chemical space is reportedly up to 10^{180} species[28, 29], which is significantly larger than the chemical subspace covered by current databases such as PubChem (123 million), Beilstein (10.3 million), and GDB-17 (167 billion). Consequently, solely relying on existing data may not lead to the yield the optimal choices for novel candidate chemicals.

1.3. Computer-Aided Molecular Design (CAMD)

In recent years, computational methods have become a promising avenue for uncovering superior chemicals in unexplored regions of chemical and property space. Computer-aided molecular design (CAMD) techniques[30, 31] refer to the algorithms capable of finding promising (yet unknown) chemicals for particular uses, by continually

doi:10.6342/NTU202403528



generating novel chemical species and testing their performance properties. However, it should be noted that the computational approach is by no means a thorough replacement for experimentation at the current stage. Since a molecular model inevitably involves simplifications, experimental validations are always irreplaceable. What it can do is *complement* experimental approaches and provide additional strategic flexibility for the development of new specialty chemicals. When trial-and-error experiments are too costly and time-consuming, CAMD can be used for preliminary evaluations. Additionally, customized databases can be built or expanded to accompany each CAMD task, which is useful for exploring new chemical knowledge and complementing the data scarcity in data-driven AI research. To demonstrate how CAMD works, we shall start with illustrating the correlation between a molecular structure and its corresponding properties.

1.3.1. Bidirectional Relation: Molecular Structure and Properties

The understanding of how molecular structure dictates a molecule's properties has been a cornerstone of molecular science. Traditionally, the focus has been on predicting properties based on a known molecular structure. However, the development of new chemicals is inherently driven by desired properties. In this context, candidate chemicals are identified based on whether they meet these predefined property requirements. This essentially represents a reverse engineering approach to property prediction, highlighting the core purpose of Computer-Aided Molecular Design (CAMD).

The aforementioned bidirectional relationship between structure and properties is depicted in **Figure 1.3-1**. In particular, property predictions for a given structure are termed the **forward algorithm**, as elaborated in section 2.3.2. The subroutine for generating new molecules and selecting better-suited ones from them is termed the **reverse algorithm**, as elaborated in section 2.3.3 and 2.3.4. As mentioned in section 1.1,

doi:10.6342/NTU202403528

CAMD techniques can be invoked to complement the probably inefficiency in traditional development process.

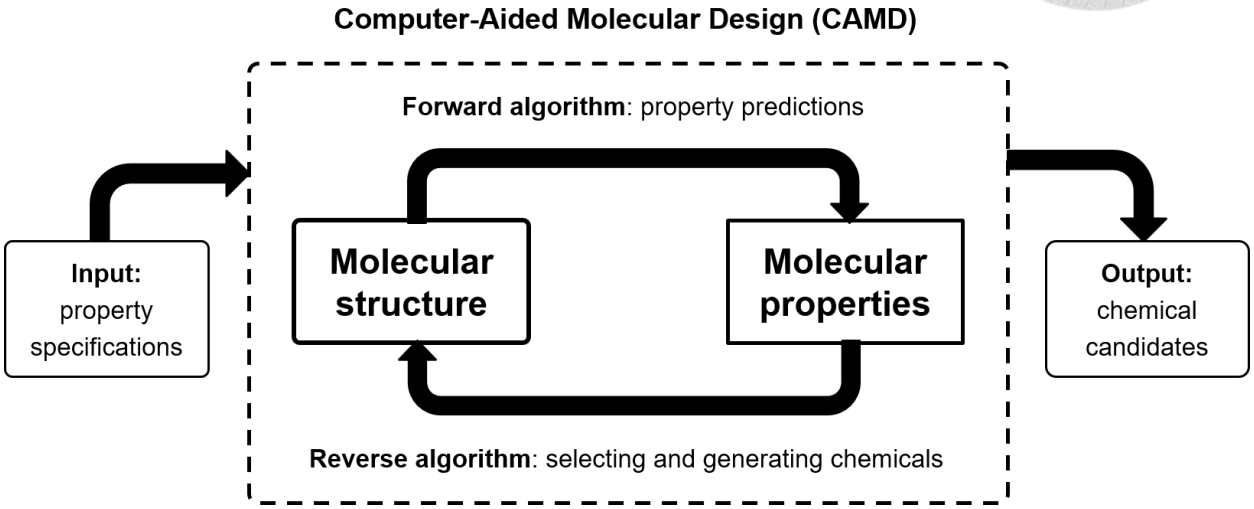


Figure 1.3-1. The bidirectional relationships between molecular structure and molecular properties: forward algorithm vs. reverse algorithm.

It should be emphasized that a CAMD task also involves predicting properties. Therefore, knowledge about the advantages and disadvantages of various predicting methods is as crucial as knowledge about the reverse algorithm. For optimization-based CAMD, a collection of optimal chemical species is generated through alternating iterations of **forward algorithm** and **reverse algorithm** stages. Specifically, a population of chemical species is initialized, followed by property predictions for these species (**forward algorithm**). The **reverse algorithm** then selects better-suited species from the population and generates novel species by modifying the selected chemical structures. The physicochemical properties of these novel species are evaluated using the **forward algorithm**, and the better-suited novel species are selected by the **reverse algorithm** for the next iteration. The process iterates until a sufficient number of chemical candidates

doi:10.6342/NTU202403528

have been found or specific termination criteria are satisfied.

Currently, there are two main methodologies in the field of molecular design: traditional approaches[32] and machine learning (ML)-based approaches[33, 34]. This work focuses mainly on the traditional approaches, whereas in Chapter 6 several ML-based approaches are reviewed and compared with traditional approaches. Depending on users' objectives, models of both types the can be employed for either exploitation or exploration tasks[35, 36]. In general, an exploitation task involves the purposeful generation of chemicals to meet predefined physicochemical property criteria, which is rightly the primary objective of a CAMD program as described earlier. In contrast, an exploration task generates chemicals without being bound by property requirements and other constraints, which is more precisely referred to as "chemical space exploration" or "molecular generation".

1.3.2. Mathematical Formulation of a CAMD Problem

As mentioned in section 1.3.1, CAMD is the reverse engineering of property predictions. Therefore, finding the inverse function of property models would be a direct solution to a CAMD problem. Unfortunately, obtaining this inverse function in analytical form can be challenging since many property prediction methods, such as quantum mechanical calculations (QM) and molecular dynamics/Monte Carlo simulations (MD/MC), only provide numerical functions in practice. Moreover, a property model can exhibit high nonlinearity in relation to chemical structures. Since a structure is often represented by multiple discrete variables (see section 2.3.1), finding the inverse function of a property model can still be difficult even if the analytical form of the property model is known. It turns out that one often needs to seek alternative ways to solve CAMD problems.

doi:10.6342/NTU202403528

From a mathematical optimization perspective, CAMD problems are typically classified as mixed-integer nonlinear programming (MINLP) problems[37]. This classification arises due to the combined presence of discrete representations for a chemical structure and the nonlinearity of property models. A chemical structure is often a structured data with some discrete properties. For instance, there should only be a few reasonable bond order types (i.e. single, double, triple, etc.) in a usual chemical structure. The theoretical details of MINLP problem are elaborated in section 2.1 and 2.2, and only its generic mathematical form is mentioned here:

Problem MINLP

$$\underset{\boldsymbol{u},\boldsymbol{w}}{\operatorname{argmin}}\ Objfcn(\boldsymbol{u},\boldsymbol{w}) \tag{1.3–1}$$

subjected to

$$\boldsymbol{h}(\boldsymbol{u},\boldsymbol{w}) = 0 \tag{1.3–2}$$

$$\boldsymbol{g}(\boldsymbol{u},\boldsymbol{w}) \leq 0 \tag{1.3–3}$$

$$\boldsymbol{u} \in U \subseteq \mathbb{Z}^n \tag{1.3–4}$$

$$\boldsymbol{w} \in W \subseteq \mathbb{R}^m \tag{1.3–5}$$

Here, $Objfcn(\boldsymbol{u},\boldsymbol{w})$ represents the objective function, $\boldsymbol{u}$ is a n-dimensional column vector of integer variables, $\boldsymbol{w}$ is a m-dimensional vector of continuous variables, $\boldsymbol{h}(\boldsymbol{u},\boldsymbol{w}) = 0$ is a vector representing p equality constraints, $\boldsymbol{g}(\boldsymbol{u},\boldsymbol{w}) \leq 0$ is a column vector representing q inequality constraints, and argmin indicates the arguments $(\boldsymbol{u}^*,\boldsymbol{w}^*)$ that minimize the value of the objective function, $Objfcn(\boldsymbol{u}^*,\boldsymbol{w}^*)$. By analogy with a molecular design task, $\boldsymbol{u}$ represents molecular structures, $\boldsymbol{w}$ typically represents thermodynamic state variables, such as pressure, temperature, and compositions. Each

doi:10.6342/NTU202403528

entry in $\boldsymbol{h}(\boldsymbol{u},\boldsymbol{w})$ or $\boldsymbol{g}(\boldsymbol{u},\boldsymbol{w})$ vector specifies certain molecular property (e.g. vapor pressure, density, viscosity etc.) or an operating constraint of the process (e.g. total annual cost, maximum allowable heat duty, etc.).

A pioneering research study on solving the practical CAMD problem was conducted by R. Gani and E. A. Brignole in 1983[38]. They demonstrated the concept that a great variety of new chemical structures can be assembled from a few molecular fragments, and that multiple optimal chemicals can be found within the chemical space spanned by these molecular fragments. In their study, a set of common functional group fragments, such as -CH2-, -CH3-, -OH, -CH2CO-, -CH2COO-, and -CH2CN, was pre-defined as the building blocks. They then used the exhaustive combinatorial enumeration to connect these functional groups in all of the possible ways, resulting in numerous new molecules. Next, they employed the UNIFAC model to predict the performance of these new molecules in the separation of aromatic mixtures. Finally, a subset of molecules that exhibit the highest performance was chosen as potential solvents for the extraction process.

Such prototype has been consistently systematized, diversified, and generalized by various research groups. To this day, the methodologies for CAMD have become a knowledge system[39-44]. Currently, the focus of molecular design research is primarily on the development of novel organic solvents, specialized ionic liquids, small-molecule drugs, and polymers, as summarized in **Table A3**.

1.4. The Purposes of This Work and an Outline

After reviewing the literature on Computer-Aided Molecular Design (CAMD), it becomes evident that many studies, particularly early ones, present challenges for outsiders aiming to apply these computational approaches directly to new applications.

doi:10.6342/NTU202403528

These difficulties arise from several factors:

- **Limited transparency of computational toolkits**

  The majority of early studies on generative algorithms[45-47] did not disclose their computer program source code. In addition, the applicability window of property prediction models may not be reported explicitly.[48]

- **Exclusiveness of computational toolkits for particular topics**

  Numerous studies focused heavily on designing small drug-like molecules, with property prediction models emphasizing metabolic properties, toxicity, and binding affinity to specific biological targets[46, 49, 50] For non-biological specialty chemicals, these properties may not be the primary considerations. In addition, a correlative property prediction model is typically only suitable for pure chemical species and limited scope of chemical mixtures.

- **Potential issues with molecular representation and complexity**

  In some early works employing group contribution (GC) methods or quantitative structure-property relationships (QSPR) for property predictions, the frequency distribution of intra-molecular features (rather than rigorous molecular connectivity) is used to represent a molecular species in their CAMD task.[38, 51-56] However, a specific frequency distribution of intra-molecular features can correspond to multiple constitutional isomers, and the GC or QSPR models may not distinguish between them.[54, 56, 57] To maintain the scale compatibility between chemical representation with GC (or QSPR) models, molecular building blocks and structure modifications are often restricted to functional groups recognizable by these models. The design at

doi:10.6342/NTU202403528

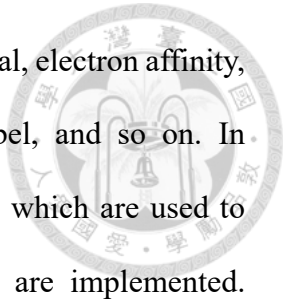
the functional group or more macroscopic level is often claimed as the "rational design"[58-61] because it rules out some unexpected substructures causing by atomic-level molecular modifications. However, it may limit the discovery of more optimal chemical species.

To investigate the capability of CAMD techniques in the design of general chemical mixtures, a customized program is built in this work for the computational mixture design at miscellaneous (i.e. atomic and fragmental) levels. A comprehensive understanding of the theoretical backgrounds of this work includes two aspects: (a) theory of mathematical optimization and (b) possible methods for the implementation of a CAMD program.

The generic theory of the two aspects is presented in Chapter 2. In section 2.1 and 2.2, the discussions include the mathematical method to solve mixed-integer non-linear programming problem (MINLP) and the difficulties for problems involving more complicated discrete variables. In section 2.3, a literature review on the implementations of a CAMD program is provided. In particular, there are four vital components in a CAMD program: (b1) molecular representations, (b2) property prediction models (forward algorithm), (b3) generative algorithm (reverse algorithm I), (b4) selection algorithms (reverse algorithm II).

Following (b1) to (b4), the implementation details of our own CAMD program are presented in Chapter 3. In particular, we develop a MARS+ package (Molecular Assembling and Representation Suite - Plus), composed of a digital representation of chemical mixtures (section 3.1) and a collection of genetic algorithm-based operators as the generative algorithm (section 3.3). Any chemical species in the format of such digital representation can be subjected to genetic operators to forming a new species. In section 3.2, the theories of the property prediction models in our program are introduced,

doi:10.6342/NTU202403528



including COSMO-SAC activity coefficient model, ionization potential, electron affinity, chemical hardness, electrophilicity, descriptors in RDKit/OpenBabel, and so on. In section 3.4, the molecular fitness function and selection algorithms, which are used to identifying better chemical species, Multiple selection algorithms are implemented. including roulette wheel (RW), simulated annealing (SA), fitness Monte Carlo (FITMC), and non-dominated sorting genetic algorithm II (NSGA-II).

Chapter 4 provides a preliminary evaluation of the intrinsic performance of MARS+ based CAMD. In section 4.1 we demonstrate the possibility of applying genetic operators to every allowable substructure in a molecular structure. In section 4.2 and 4.3, we demonstrate the possibility that MARS+ can cover sufficiently large chemical space and produce well-known molecules.

Chapter 5 exemplifies the MARS+ based CAMD in the design of novel ionic liquids (ILs) as the carbon dioxide absorbents. Section 5.1 reviews the mechanistic studies on IL-based carbon capture and storage (CCS) techniques. Section 5.2 and 5.3 presents the thermodynamic modeling for predicting CO_2 solubility in ILs and the accuracy of prediction employing COSMO-SAC activity coefficient model. Section 5.4 and 5.6 present the results of designed ILs, which indicate that the component-screening method can expand our knowledge scope from experimentally validated ILs, and that the CAMD techniques can further improve the knowledge of component-screening method.

In Chapter 6 we use two tailor-made benchmarks, GuacalMol and MolOpt, to compare the performance differences between our CAMD program and other baseline models (including AI-based and conventional ones) in some specially-devised tasks. Section 6.1 and Appendix F cover the theoretical backgrounds of some AI models. Section 6.2 shows the results of comparisons. It verifies that the performance of our program is better than many AI-based generative models and comparable to most of rule-

doi:10.6342/NTU202403528

based models.

Finally, we summarize the insights gained from the aforementioned works in Chapter 7, and provides some prospects in Chapter 8. Several potential topics that can be the extension of this work, including the design of special chemical systems such as co-crystals, and the incorporation with process design techniques to form a computer-aided molecular-process design (CAMPD) scheme.

doi:10.6342/NTU202403528

Chapter 2. Generic Theory

2.1. Mixed-Integer Non-Linear Programming (MINLP)

As revealed in section 1.3.2, a CAMD task can be framed as a MINLP problem. In this section, more specifications are added to generic MINLP form (section 1.3.2) to adapt it into the real implementation in this work. Based on Gibbs phase rule[62], the equilibrium thermodynamic state $\boldsymbol{s_i}$ of a $\mathcal{C}$-component mixture system in a phase can be described by $\boldsymbol{s_i}(\boldsymbol{u_i}.\boldsymbol{w_i})$, where $\boldsymbol{u_i}$ is the mixture species and $\boldsymbol{w_i}$ are $(\mathcal{C}+1)$ intensive variables. In particular, it is common to let $\boldsymbol{w_i}$ represent temperature, pressure and mole fractions (i.e. $\boldsymbol{w_i} = [T; P; x_1; \dots; x_{\mathcal{C}-1}]$), as many processes are carried out under isobaric-isothermal condition. Therefore, $\boldsymbol{w_i}$ is usually continuous. The mixture can be denoted as $\boldsymbol{u_i} = [\boldsymbol{u_{i1}}; \dots; \boldsymbol{u_{ic}}]$, where each $\boldsymbol{u_{ij}}$ is a pure chemical component. Since each $\boldsymbol{u_{ij}}$ in mixture is typically expressed in certain chemoinformatic format[63] of chemical structure, it usually exhibits discrete properties. For instances, the number of constituent atoms in a molecule must be an integer, and the feasible bond orders must be one of a few discrete options, e.g. single, double, triple bonds. In terms of these arguments, the CAMD task formulated as the following form[32, 37]:

Problem CAMDMINLP

$$\underset{\mathbf{u_i},\mathbf{w_i}}{\operatorname{argmin}}\ Objfcn(\boldsymbol{u_i}, \boldsymbol{w_i}; \boldsymbol{t^h}, \boldsymbol{t^g}) \tag{2.1–1}$$

subjected to

$$\boldsymbol{h}(\boldsymbol{z}(\boldsymbol{u_i}), \boldsymbol{w_i}) = 0 \tag{2.1–2}$$

$$\boldsymbol{g}(\boldsymbol{z}(\boldsymbol{u_i}), \boldsymbol{w_i}) \leq 0 \tag{2.1–3}$$

doi:10.6342/NTU202403528

$$\boldsymbol{e}(\boldsymbol{u_i}) = 0 \tag{2.1–4}$$

$$\mathbf{u_i} \in \mathbb{Z}^{a\times b} \subseteq \mathbf{X} \subseteq \mathbb{U} \tag{2.1–5}$$

$$\mathbf{w_i} \in \mathbb{R}^{\mathcal{C}+1}, \mathbf{w_i^L} \leq \mathbf{w_i} \leq \mathbf{w_i^U} \tag{2.1–6}$$

Here, $\boldsymbol{z}$ stands for generic chemoinformatic software that can convert among different chemoinformatic formats of molecular structure, e.g. RDKit[64] and OpenBabel[65]. $\boldsymbol{h}(\boldsymbol{z}(\boldsymbol{u_i}), \boldsymbol{w_i}) = \boldsymbol{f^h}(\boldsymbol{z}(\boldsymbol{u_i}), \boldsymbol{w_i}) - \boldsymbol{t^h} = 0$ represents p equality constraints on molecular properties, where $\boldsymbol{f^h}(\boldsymbol{z}(\boldsymbol{u_i}), \boldsymbol{w_i}) = [f_1(\boldsymbol{z}(\boldsymbol{u_i}), \boldsymbol{w_i}); \dots; f_p(\boldsymbol{z}(\boldsymbol{u_i}), \boldsymbol{w_i})]$ are the models or methodologies for property estimations, and $\boldsymbol{t^h} = [t_1; \dots; t_p]$ are the target values of properties for equality constraints. Similarly, $\boldsymbol{g}(\boldsymbol{z}(\boldsymbol{u_i}), \boldsymbol{w_i}) = \boldsymbol{t^g} - \boldsymbol{f^g}(\boldsymbol{z}(\boldsymbol{u_i}), \boldsymbol{w_i}) \leq 0$ represents q inequality constraints on molecular properties, where $\boldsymbol{f^g}(\boldsymbol{z}(\boldsymbol{u_i}), \boldsymbol{w_i}) = [f_{p+1}(\boldsymbol{z}(\boldsymbol{u_i}), \boldsymbol{w_i}); \dots; f_{p+q}(\boldsymbol{z}(\boldsymbol{u_i}), \boldsymbol{w_i})]$ are the models or methodologies for property estimations, and $\boldsymbol{t^g} = [t_{p+1}; \dots; t_{p+q}]$ are the target boundaries for the values of property. $\boldsymbol{e}(\boldsymbol{u_i}) = 0$ represents a filter function for chemical structure $\boldsymbol{u_i}$. It can be devised to preserve particular molecular structural features, or to impose an intended bias, during optimization process. It essentially belongs to one of the equality constraints $\boldsymbol{h}(\boldsymbol{z}(\boldsymbol{u_i}), \boldsymbol{w_i}) = 0$, but is written explicitly here for introduction. $Objfcn(\boldsymbol{u_i}, \boldsymbol{w_i}; \boldsymbol{t^h}, \boldsymbol{t^g})$ is the objective function that determines the optimality of species $\boldsymbol{u_i}$. In this work, it is replaced with a fitness function $Fitfcn(\boldsymbol{u_i}, \boldsymbol{w_i}; \boldsymbol{t^h}, \boldsymbol{t^g})$ so as to align with the framework of genetic algorithm. The maximization of $Fitfcn(\boldsymbol{u_i}, \boldsymbol{w_i}; \boldsymbol{t^h}, \boldsymbol{t^g})$ is equivalent to the minimization of $Objfcn(\boldsymbol{u_i}, \boldsymbol{w_i}; \boldsymbol{t^h}, \boldsymbol{t^g})$.

The $\mathbb{U}$ in eq (2.1–5) denotes the chemical space[25, 28], a set of all the theoretically feasible chemicals typically subjected to the expanded octet rule[66, 67] and intrinsic constraints from the chemical representation in use. For the numerical stability in

doi:10.6342/NTU202403528

optimization and the feasibility of chemical utilization, it is desirable that all the temporary solutions (i.e. the chemical mixtures $\boldsymbol{u_i} = [\mathbf{u_{i1}}, \dots, \mathbf{u_{ic}}]$) during the optimization process is a subset of the chemical space. The $\boldsymbol{w_i^L}$ and $\boldsymbol{w_i^U}$ in eq (2.1–6) are the lower and the higher boundaries to define the feasible region of $\boldsymbol{w_i}$. Finally, the overall optimality of a designed species $\mathbf{u_i}$ at the associated thermodynamic state $\boldsymbol{w_i}$ is determined by objective function $Objfcn(\boldsymbol{u_i}, \boldsymbol{w_i}; \boldsymbol{t^h}, \boldsymbol{t^g})$. After the optimization of the objective function fulfills convergence criteria, a collection of optimal $(\boldsymbol{u_i}, \boldsymbol{w_i})$ sets will be reported. It is noteworthy that, in practice, the optimization typically starts with multiple initial solutions $\mathbf{X} = \{\mathbf{u_1}, \dots, \mathbf{u_N}\}$, also known as the "population" in genetic algorithm. All the solutions are updated simultaneously in a single optimization step.

2.2. Mathematical Methods for Solving MINLP Problems

To make it simple, this section follows the problem and notations introduced in Problem MINLP (section 1.3.2). The level of difficulty in solving a MINLP problem lies in the characteristics of the involved functions, including $Objfcn(\boldsymbol{u}, \boldsymbol{w})$, $\boldsymbol{h}(\boldsymbol{u}, \boldsymbol{w})$, and $\boldsymbol{g}(\boldsymbol{u}, \boldsymbol{w})$. In this section, two scenarios are discussed.

In the first scenario, $\boldsymbol{u}$ represents several independent integer variables, and all the functions are continuously differentiable for every of the function arguments. In other words, $Objfcn(\boldsymbol{u}, \boldsymbol{w})$, $\boldsymbol{h}(\boldsymbol{u}, \boldsymbol{w})$, and $\boldsymbol{g}(\boldsymbol{u}, \boldsymbol{w})$ are well-defined, continuous, and differentiable at $\boldsymbol{u}$ even $\boldsymbol{u}$ is a non-integer real number. This case is presented in section 2.2.1.

In the second scenario, $\boldsymbol{u}$ represents complicated structured data, and all the functions are well-defined only at particular discrete $\boldsymbol{u} \in \{\boldsymbol{u_1}, \boldsymbol{u_2}, \dots\}$. Such scenario makes the problem difficult to solve for $\boldsymbol{u}$ by utilizing derivative and continuity. This

doi:10.6342/NTU202403528

case is presented in section 2.2.2.

2.2.1. Continuously Differentiable Problem with Integer Variables

This category of optimization problem can be tackled by using branch-and-bound (BB) method[68]. Harnessed with the method, one can systemically decompose the original optimization problem into several subproblems on different feasible regions. To start with, let us consider a simple linear programming problem that is solvable by graphical method:

Problem CDMINLP-P1 (continuously differentiable MINLP, problem 1)

$$\max_{x_1,x_2} j(x_1,x_2) = 3x_1 + 4x_2 \quad (2.2\text{–}1)$$

subjected to

$$7x_1 + 11x_2 \leq 88 \quad (2.2\text{–}2)$$

$$3x_1 - x_2 \leq 12 \quad (2.2\text{–}3)$$

$$x_1 \geq 0, x_1 \in \mathbb{Z} \quad (2.2\text{–}4)$$

$$x_2 \geq 0, x_2 \in \mathbb{Z} \quad (2.2\text{–}5)$$

In the first step, Problem CDMINLP-P1 is solved by treating x_1 and x_2 as continuous variables. The optimal continuous solution is found to be $\boldsymbol{x_{P1}^*} = [5.5, 4.5]^{\boldsymbol{T}}$ with $f(\boldsymbol{x_{P1}^*}) = 34.5$. Based on solution $\boldsymbol{x_{P1}^*}$, one can either *branch* $x_1 \geq 6$ and $x_1 \leq 5$ for variable x_1, or $x_2 \geq 5$ and $x_2 \leq 4$ for variable x_2. Suppose x_1 is chosen for branching, then Subproblem CDMINLP-P2 and Subproblem CDMINLP-P3 are generated:

Subproblem CDMINLP-P2

doi:10.6342/NTU202403528

A duplicate of Problem CDMINLP-P1 with eq (2.2–4) changed into eq (2.2–6).

$$x_1 \leq 5, x_1 \in \mathbb{Z} \quad (2.2\text{–}6)$$

Subproblem CDMINLP-P3

A duplicate of Problem CDMINLP-P1 with eq (2.2–4) changed into eq (2.2–7).

$$x_1 \geq 6, x_1 \in \mathbb{Z} \quad (2.2\text{–}7)$$

Again, Subproblem CDMINLP-P2 is solved by treating x_1 and x_2 as continuous variables, and the optimal continuous solution is $\boldsymbol{x}_{\boldsymbol{P2}}^{*} = [5, 4.8]^{\boldsymbol{T}}$ with $f(\boldsymbol{x}_{\boldsymbol{P2}}^{*}) = 34.3$. On the other hand, Subproblem CDMINLP-P3 has no feasible solution. Next, *branching* $x_2 \geq 5$ and $x_2 \leq 4$ from Subproblem CDMINLP-P2 based on solution $\boldsymbol{x}_{\boldsymbol{P2}}^{*}$. Subproblem CDMINLP-P4 and Subproblem CDMINLP-P5 are generated.

Subproblem CDMINLP-P4

A duplicate of Problem CDMINLP-P1 with eq (2.2–4) and eq (2.2–5) changed into eq (2.2–8) and eq (2.2–9).

$$x_1 \leq 5, x_1 \in \mathbb{Z} \quad (2.2\text{–}8)$$

$$x_2 \leq 4, x_2 \in \mathbb{Z} \quad (2.2\text{–}9)$$

Subproblem CDMINLP-P5

A duplicate of Problem CDMINLP-P1 with eq (2.2–4) and eq (2.2–5) changed into eq (2.2–10) and eq (2.2–11).

$$x_1 \leq 5, x_1 \in \mathbb{Z} \quad (2.2\text{–}10)$$

$$x_2 \geq 5, x_2 \in \mathbb{Z} \quad (2.2\text{–}11)$$

doi:10.6342/NTU202403528

Subproblem CDMINLP-P4 is solved by treating x_1 and x_2 as continuous variables, and the optimal continuous solution is $\boldsymbol{x}_{\boldsymbol{P4}}^{*} = [5, 4]^{\boldsymbol{T}}$ with $f(\boldsymbol{x}_{\boldsymbol{P4}}^{*}) = 31$. Subproblem CDMINLP-P5 is solved in the same manner, resulting in $\boldsymbol{x}_{\boldsymbol{P5}}^{*} = [0, 8]$ with $f(\boldsymbol{x}_{\boldsymbol{P5}}^{*}) = 32$. Since the solution to each of the two subproblems happens to be integers, one no longer needs to examine integer solutions in the vicinity of the continuous solution. Consequently, $\boldsymbol{x}_{\boldsymbol{P5}}^{*}$ is determined as the optimal solution to Problem CDMINLP-P1. This overall strategy can be represented by the following decision-tree diagram.

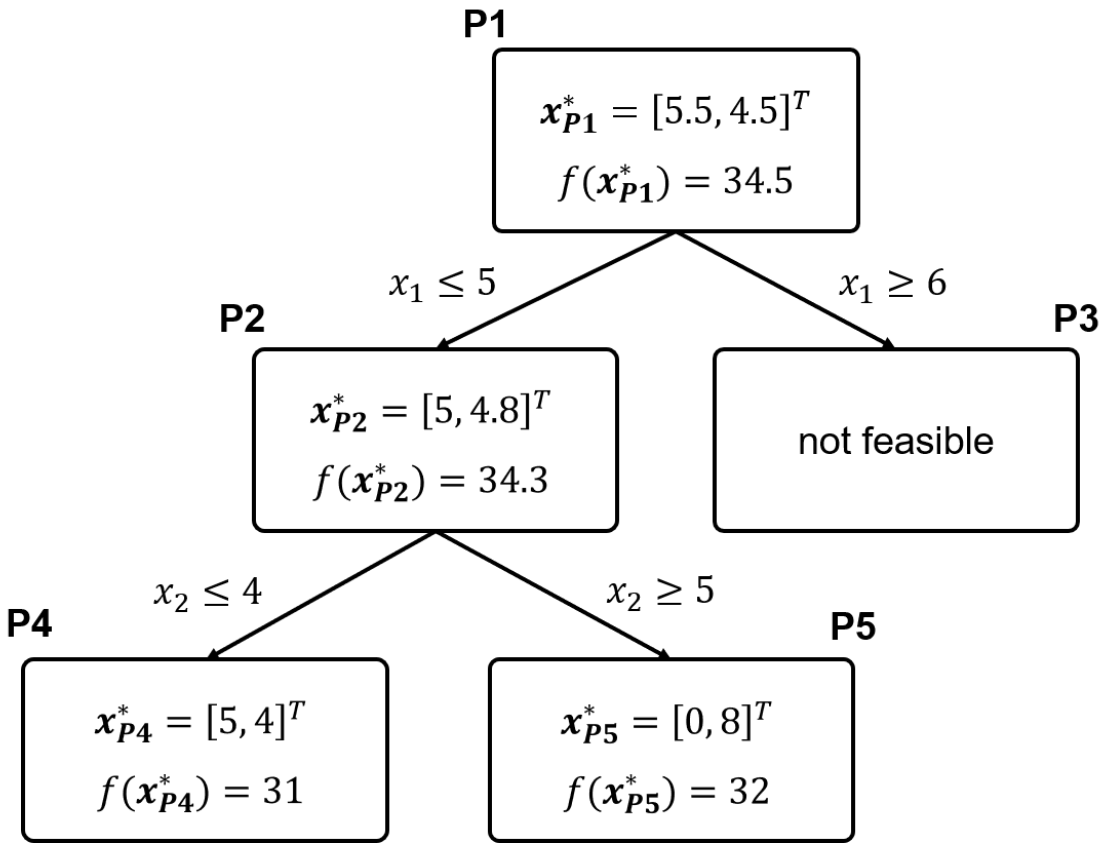


Figure 2.2-1. The branch-and-bound process for solving CDMINLP-P1.

If eqs (2.2–1) to (2.2–3) are nonlinear functions, one can resort to nonlinear optimization methods, such as sequential quadratic programming (SQP, see Appendix D).

doi:10.6342/NTU202403528

2.2.2. Nondifferentiable Problem with Complicated Discrete Variables

One of the most general methods to this category of problems is generalized benders decomposition (GBD)[37, 69, 70]. Its theoretical formulation is provided in Appendix E. The central idea of GBD is to decompose MINLP problem into a *nonlinear programming* (NLP) task and an *integer nonlinear programming* (INLP) task. In particular, the former one is known as the *primal problem*, and the latter known as the *master problem*. Following the notation in Problem MINLP (section 1.3.2), the details of GBD is introduced below. Let $\boldsymbol{u_i}$ and $\boldsymbol{w_i}$ denote the values of the discrete and continuous variables, respectively, where the subscript i is meant to differentiate among the discovered feasible solutions during optimization. GBD algorithm starts with substituting initial-guess values $\boldsymbol{u_1}$ for discrete variables $\boldsymbol{u}$, making the MINLP problem reduce to a nonlinear programming (NLP) *primal* problem:

Problem GBD-P-NLP($\boldsymbol{u_1}$) (GBD primal problem, nonlinear programming)

$$\min_{\boldsymbol{w}} Objfcn(\boldsymbol{u_1}, \boldsymbol{w}) \tag{2.2–12}$$

subjected to

$$\boldsymbol{h}(\boldsymbol{u_1}, \boldsymbol{w}) = 0 \tag{2.2–13}$$

$$\boldsymbol{g}(\boldsymbol{u_1}, \boldsymbol{w}) \leq 0 \tag{2.2–14}$$

$$\boldsymbol{w} \in W \subseteq \mathbb{R}^m \tag{2.2–15}$$

Set $k = 1$ to indicate current subscript of $\boldsymbol{u_1}$ and set counter $r = 0$ to counts the infeasible optimal solution $\boldsymbol{w}^*$ in the *master problem* (detailed in next paragraphs). By solving the *primal problem* using nonlinear optimization methods (Appendix D), GBD

doi:10.6342/NTU202403528

algorithm obtains an optimal solution $\boldsymbol{w_1}$ for the continuous variables $\boldsymbol{w}$, as well as the Lagrange multiplier vectors ($\boldsymbol{\lambda_1}, \boldsymbol{\mu_1}$) corresponding to constraints $\boldsymbol{h}(\boldsymbol{u_1}, \boldsymbol{w_1})$ and $\boldsymbol{g}(\boldsymbol{u_1}, \boldsymbol{w_1})$, respectively. The value of objective function obtained at this stage, $Objfcn(\boldsymbol{u_1}, \boldsymbol{w_1})$, is referred to as the *upper bound* Z_U in GBD algorithm. With $\boldsymbol{w_1}$ known, the *master problem* is subsequently formulated.

Problem GBD-M-INLP($\boldsymbol{w_t}$) (GBD master problem, integer nonlinear programming)

$$\min_{\alpha \in \mathbb{R}, \boldsymbol{u}} \ \alpha \tag{2.2–16}$$

subjected to

$$\alpha \geq Objfcn(\boldsymbol{u}, \boldsymbol{w_t}) + \sum_{i=1}^{p} (\lambda_t)_i h_i(\boldsymbol{u}, \boldsymbol{w_t}) + \sum_{j=1}^{q} (\mu_t)_j g_j(\boldsymbol{u}, \boldsymbol{w_t}), \quad t = 1, \dots, k \tag{2.2–17}$$

$$\boldsymbol{u} \in V \tag{2.2–18}$$

$$\boldsymbol{\mu} \geq \boldsymbol{0} \tag{2.2–19}$$

$$\boldsymbol{\lambda} \in \mathbb{R}^p \tag{2.2–20}$$

$$\boldsymbol{\mu} \in \mathbb{R}^q \tag{2.2–21}$$

with $V = \{\boldsymbol{u} \,|\boldsymbol{h}(\boldsymbol{u}, \boldsymbol{w_t}) = 0, \boldsymbol{g}(\boldsymbol{u}, \boldsymbol{w_t}) \leq 0 \text{ for } \boldsymbol{w_t}, t = 1, \dots, r\}$

Solving the *master problem* yields optimal solutions $\boldsymbol{u}^*$ and α^*. Notably, α^* is designated as the *lower bound* Z_L in GBD algorithm. This designation arises because α^* serves as a lower bound for the objective function, $Objfcn(\boldsymbol{u}, \boldsymbol{w})$, in the Problem MINLP (see Appendix E.3). If $Z_L \geq Z_U$, then the solution $(\boldsymbol{u}^*, \boldsymbol{w_t}, \boldsymbol{\lambda_t}, \boldsymbol{\mu_t})$ corresponding to α^* is the optimal solution (see Appendix C.6 and E.3). If not, solve the *primal problem* with $\boldsymbol{u}$ fixed at $\boldsymbol{u}^*$. Obtain an optimal solution $\boldsymbol{w}^*$ for the continuous variables $\boldsymbol{w}$, as

doi:10.6342/NTU202403528

well as Lagrange multiplier vectors $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ corresponding to constraints $\boldsymbol{h}(\boldsymbol{u}^*, \boldsymbol{w}^*)$ and $\boldsymbol{g}(\boldsymbol{u}^*, \boldsymbol{w}^*)$, respectively. If optimal solution $\boldsymbol{w}^*$ is infeasible, include $\boldsymbol{w}^*$ in V. Set $r = r + 1$, $\boldsymbol{w_r} = \boldsymbol{w}^*$. Return to the *master problem*.

If $Objfcn(\boldsymbol{u}^*, \boldsymbol{w}^*) \leq Z_L$, then the solution $(\boldsymbol{u}^*, \boldsymbol{w}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ is identified as the optimal solution. Otherwise, set $k = k + 1$, $\boldsymbol{w_k} = \boldsymbol{w}^*$, and $\boldsymbol{u_k} = \boldsymbol{u}^*$. Set $Z_U = Objfcn(\boldsymbol{u}^*, \boldsymbol{w}^*)$ if $Objfcn(\boldsymbol{u}^*, \boldsymbol{w}^*)$ is less than current Z_U. Return to the *master problem*. A numerical example can be seen in reference[69]. The overall process is illustrated in **Figure 2.2-2**.

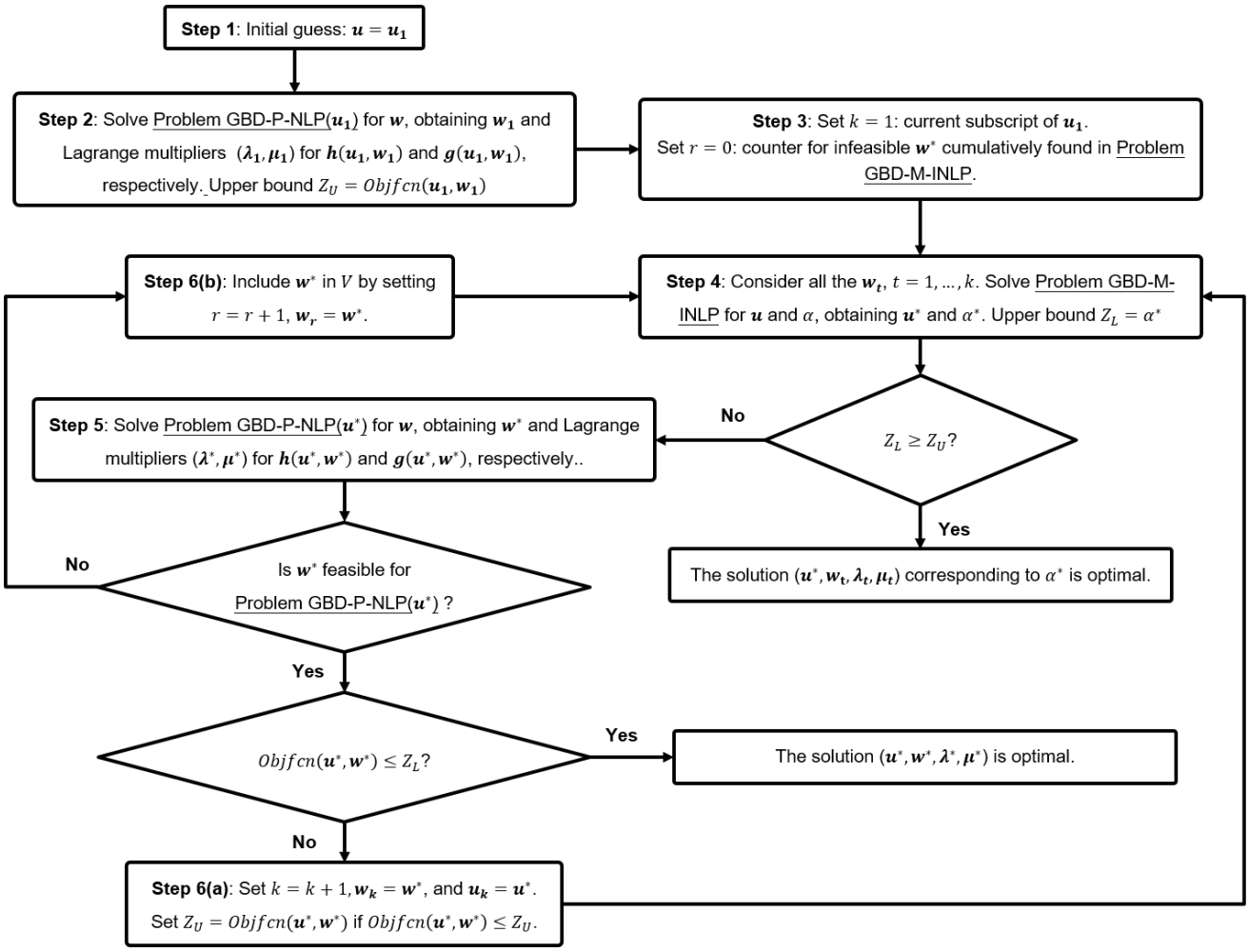


Figure 2.2-2. The flowchart of solving MINLP problem by GBD method.

doi:10.6342/NTU202403528

2.3. Components of Computer-Aided Molecular Design (CAMD)

In the framework of rule-based CAMD, there are four vital components: molecular data structure (MDS), property prediction methods, generative algorithms, selection algorithms. **Figure 2.3-1**, which is a detailed version of **Figure 1.3-1**, indicates the mathematical relations and the roles of these components.

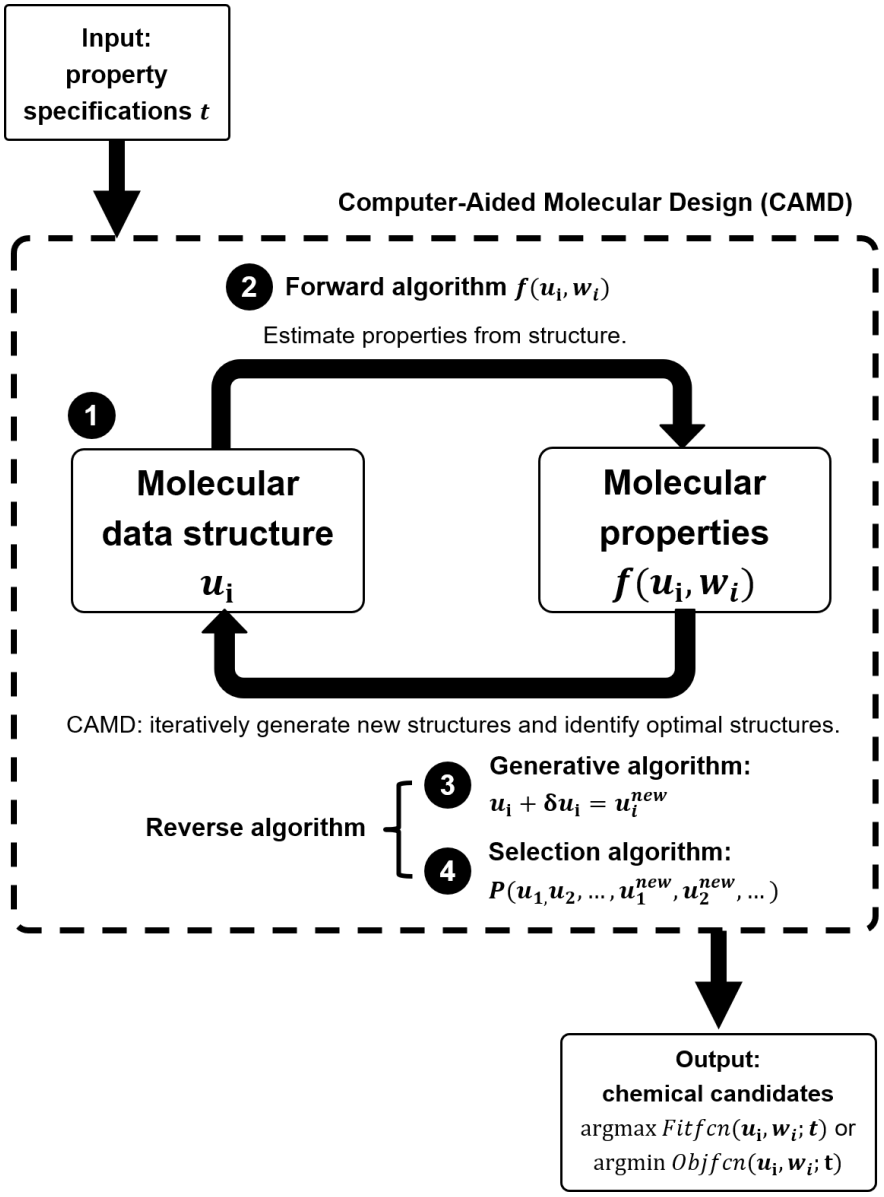


Figure 2.3-1. The four components of rule-based computer-aided molecular design.

doi:10.6342/NTU202403528

2.3.1. Molecular Data Structure: Chemical Representations

There are grossly four types of digital representation of a molecular structure[63, 71, 72] $\underline{\mathbf{x}}_i$: 1D string representation, 1D fingerprint representation, 2D graph (matrix) representation, and 3D representation, and multi-dimensional latent space representation. In particular, 1D string representation (e.g. SMILES[73, 74], InChI[75], SMARTS[73], SELFIES[76] etc.) has the advantages of compactness and readability, though it is generally not suitable for structure variations due to their syntactic complexity.

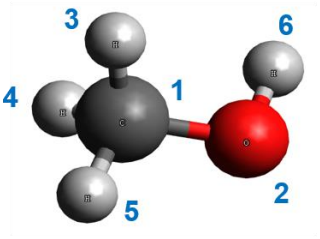
Molecular fingerprint[77-79] is an encoding system generally based on the specific traits of 1D, 2D, 3D representation, or other descriptors. For example, the MACCS[80] can track the types and the quantities of neighboring atoms for each atom within a molecule. Additionally, MACCS can record membership of atoms in specific substructures, such as rings, aromatic bonds, and C=C bonds. The original molecular structure can be recovered by putting together all the identified fingerprint features. Chemoinformatic toolkits, such as OpenBabel[81] and RDKit[82], are useful tools for providing the aforementioned rule-based molecular representations.

2D graph representation[72, 83] has clear representation for molecular connectivity, which facilitates substructure variations. However, it is usually not as readable as string representation. Both 1D and 2D representation can contain information of constituent atoms, bond orders, constitutional isomerism, cis-trans isomerism, enantiomerism, and diasteriomerism, but they often lack conformational information.

When the design task is geometry-sensitive, 3D representation[72] is usually the most suitable. For example, the bioactivities of biomolecules often depend largely on their geometric compatibility with binding sites of substrates, hence the structure-based drug design are usually based on 3D representation.[84, 85]

doi:10.6342/NTU202403528

Table 2.3-1. Exemplary representations for methanol molecule.

<table>
<tr><th>Dimension</th><th colspan="2">Representation</th></tr>
<tr><td>3D</td><td colspan="2"> 

<table>
<tr><td>C</td><td>1.09448</td><td>-0.07782</td><td>0.01634</td></tr>
<tr><td>O</td><td>2.49247</td><td>-0.04942</td><td>0.00891</td></tr>
<tr><td>H</td><td>0.71948</td><td>-0.33717</td><td>1.03013</td></tr>
<tr><td>H</td><td>0.71947</td><td>-0.80054</td><td>-0.74043</td></tr>
<tr><td>H</td><td>0.71310</td><td>0.93005</td><td>-0.24743</td></tr>
<tr><td>H</td><td>2.78485</td><td>-0.96681</td><td>0.24900</td></tr>
</table></td></tr>
<tr><td>2D</td><td colspan="2">Adjecency matrix (A)
<table>
<tr><th>A</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>2</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>3</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>4</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>5</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>6</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
Distance matrix (D)
<table>
<tr><th>D</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>2</td></tr>
<tr><td>2</td><td>1</td><td>0</td><td>2</td><td>2</td><td>2</td><td>1</td></tr>
<tr><td>3</td><td>1</td><td>2</td><td>0</td><td>2</td><td>2</td><td>3</td></tr>
<tr><td>4</td><td>1</td><td>2</td><td>2</td><td>0</td><td>2</td><td>3</td></tr>
<tr><td>5</td><td>1</td><td>2</td><td>2</td><td>2</td><td>0</td><td>3</td></tr>
<tr><td>6</td><td>2</td><td>1</td><td>3</td><td>3</td><td>3</td><td>0</td></tr>
</table>
Connectivity matrix (C)
<table>
<tr><th>C</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>2</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>3</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>4</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>5</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>6</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table></td></tr>
<tr><td rowspan="4">1D</td><td>FP2 fingerprint (Open Babel)</td><td>0 8 1 6 <515></td></tr>
<tr><td>FP4 fingerprint (Open Babel)</td><td>Alcohol C_ONS_bond</td></tr>
<tr><td>SMILES</td><td>C(O)</td></tr>
<tr><td>InChI</td><td>InChI=1S/CH4O/c1-2/h2H,1H3</td></tr>
</table>

2.3.2. Forward Algorithm: Property Predictions Methods

Numerous models and methodologies exist for predicting molecular properties for chemical structures. These include group contribution models (GC), quantitative structure-property relationships (QSPR), molecular dynamics simulations (MD), Monte Carlo simulations (MC), and ab-initio quantum mechanical calculations (QM). Selecting the appropriate method is crucial, considering the length and time scales of the physicochemical phenomena involved, as depicted in **Figure 2.3-2**.

doi:10.6342/NTU202403528

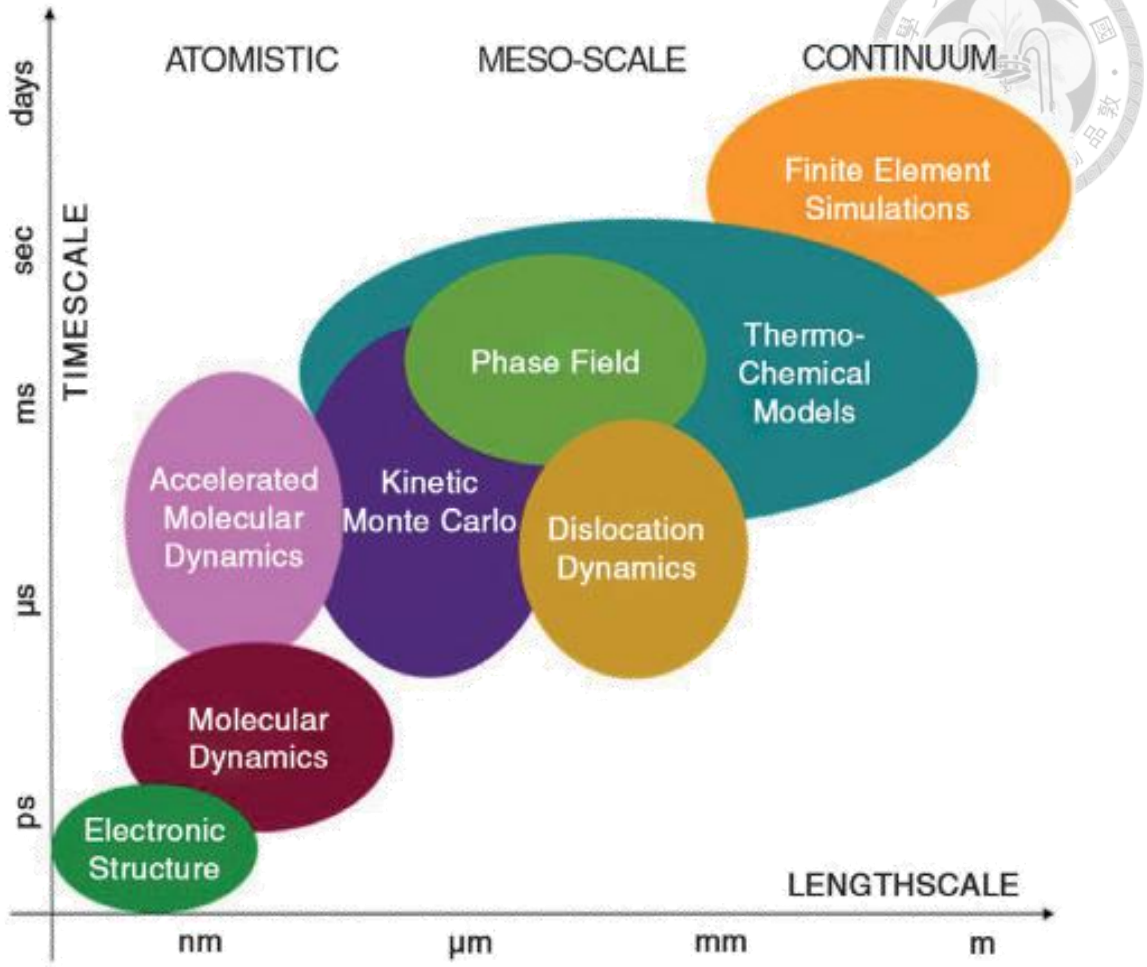


Figure 2.3-2. Each methodology for property prediction is suitable for describing physicochemical phenomena under specific scales of time and length. Reprinted with permission from the reference[86]. Copyright 2009 Elsevier Ltd.

Correlative models, such as GC and QSPR models, require substantial experimental data to regress their numerous model parameters. GC models recognize a molecule as many pre-defined functional groups in connection, and by regression, the value of a molecular property is factorized into the contribution from each of the constituent functional groups. On the other hand, the QSPR maps multiple molecular descriptors to a molecular property. Common descriptors include physicochemical properties such as the octanol/water partition coefficient (logP) and topological polar surface area (TPSA),

doi:10.6342/NTU202403528

along with structural features like the number of double bonds. Depending on the types of regression datasets, GC and QSPR models can estimate melting point, boiling point, density, viscosity, electrical conductivity, thermal conductivity, and activity coefficient. In general, these models are the highly accurate when it is used for the molecular structures similar to that in the regression datasets. However, it may be significantly inaccurate for molecules beyond the scope of regression datasets. There are also further limitations to their applicability. The group contribution model is only applicable to molecules that are composed of pre-defined functional groups. As a result, its robustness may be challenged by novel chemicals generated in computational molecular design. On the other hand, the applicability of QSPR models is usually limited to specific conditions, such as a fixed temperature.

Molecular dynamics simulations (MD) and Monte Carlo simulation (MC) are based on the theory of statistical mechanics[87]. The simulation system usually contains multiple molecules, and the interaction energy among atoms should be well-described by a proper force field. In the simulation, the system evolves continuously based on mechanical principles or sampling algorithms. As a result, the system properties, such as pressure, energy, and spatial distribution of particles, are also varying with simulation steps. After the system is equilibrated, macroscopic properties can be derived from the evolution trajectory of the system properties, via statistical mechanical interpretation. Literatures have shown that MD/MC can estimate solubility[88], transport properties[89], surface tension[90], melting and boiling points[91, 92], and free energy[87, 93]. However, the coarse graining of model[94] and parameterization of force fields[95] may require significant time and effort, Also, the computational cost of MD/MC simulation is usually higher than the correlative models.

Quantum mechanical simulations (QM) treat a molecule as a system comprising

doi:10.6342/NTU202403528

electrons and nuclei of its constituent atoms. Utilizing the Born-Oppenheimer approximation, the Schrödinger equation is formulated for the system. Once solved using methods based on Hartree-Fock theory or density functional theory (DFT), various molecular properties can be determined. These include the highest occupied molecular orbital (HOMO), lowest unoccupied molecular orbital (LUMO), ionization potential (IP), electron affinity (EA), bond order index[96], dipole moment, chemical hardness[97], electrophilicity index [97], exciton binding energy[2], enthalpy and free energy of formation[98, 99], and solvation free energy[100-102]. The QM-based thermodynamic models, such as PR+COSMOSAC and COSMO-SAC, are found useful for complementing the lack of thermodynamic parameters in process design, although their overall accuracy of predicted properties needs further improvements[103]. The primary advantage of QM methods is their reduced reliance on empirical parameters compared to correlative models and MD/MC simulations. Additionally, QM methods are universally applicable to molecular systems. However, a significant drawback is their computational cost, which escalates exponentially with the size of the molecule.

While the fundamental nature of QM methods offers generality, correlative models and coarse-grained MD/MC simulations are often more suitable choices for studying macromolecules and drug molecules due to their computational efficiency relative to QM methods. For instance, QSPR models are valuable tools in computer-aided drug design (CADD) and have demonstrably contributed to the development of commercialized drugs such as Captopril, Dorzolamide, Zanamivir, and Boceprevir.[104] MD/MC simulations find frequent application in molecular docking research for drug and material design.[104] These simulations aid in the quantitative identification of the location and strength of intermolecular interactions involving donor and acceptor sites.

doi:10.6342/NTU202403528

Table 2.3-2. A summary for the aforementioned methods for property predictions.

Method	Advantages	Disadvantages
QSPRs GC models	● A wide variety of property models can be developed once regression data is available. ● Their calculations are fast. ● They are typically accurate for molecules similar to those found in regression datasets.	● They require a substantial amount of data for regression of model parameters. ● They may exhibit significant inaccuracies for molecules outside the scope of regression datasets. ● They may have fewer theoretical foundations than MD/MC and QM.
MD/MC simulations	● They can compute a variety of thermodynamic and transport properties. ● They are grounded in the principles of classical dynamics and statistical mechanics. ● They elucidate molecular mechanisms underlying physicochemical phenomena.	● Parameterization and coarse-graining can require significant effort. ● They may necessitate substantial computational resources.
QM simulations	● They can investigate kinetic, electronic, and thermochemical properties. ● They can serve as a foundation for the development of physicochemical models ● They require significantly fewer empirical parameters than correlative models. ● In principle, they are applicable to any molecular system.	● They may require substantial computational resources. ● Computational cost scales exponentially with molecular size. This limits the applicability of QM methods to relatively small systems

doi:10.6342/NTU202403528

2.3.3. Reverse Algorithm, Part (I): Generative Algorithms

In fact, a significant focus of CAMD research has been the development of generative algorithms, particularly those specialized for designing drug-like molecules[45-47]. These algorithms can be generally categorized into three main approaches[105]: atom-based modifications, fragment-based modifications and reaction-based modifications. These approaches are illustrated in **Table 2.3-3**. Atom-based modifications excel in their ability to explore extensive chemical space through a compact set of meticulously crafted rules. Conversely, fragment-based modifications, while enhancing synthetic feasibility and reducing combinatorial complexity, come at the cost of constraining the accessible chemical space. Reaction-based modifications offer the advantages of well-established synthetic feasibility and practical synthesis pathways. However, it is crucial to point out that reaction-based modifications can introduce significant structural alterations.[105] Therefore, it is advisable to employ reaction-based modifications primarily for a rough exploration of chemical space.[105] Furthermore, the accessible chemical space from reaction-based modifications is heavily dependent upon the scope of reaction templates and the specific types of subject chemicals.

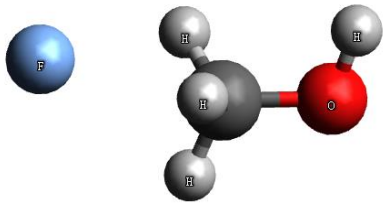
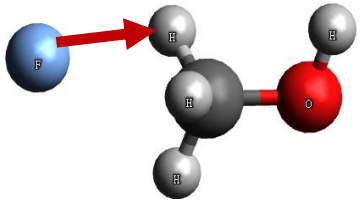
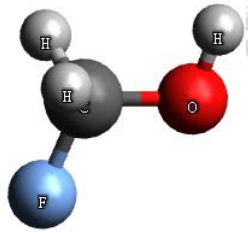
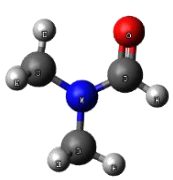
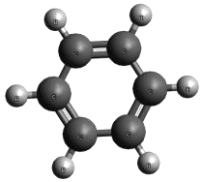
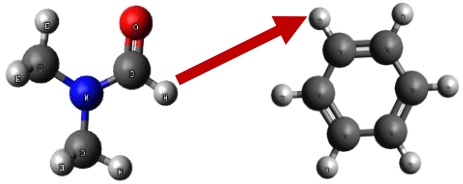
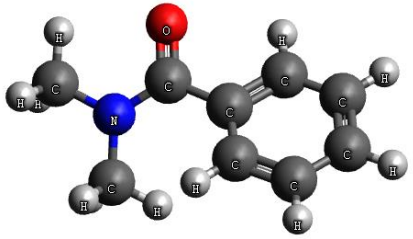
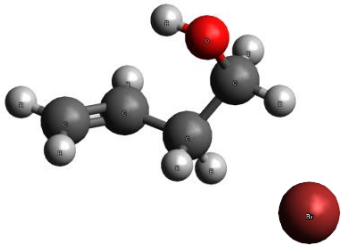
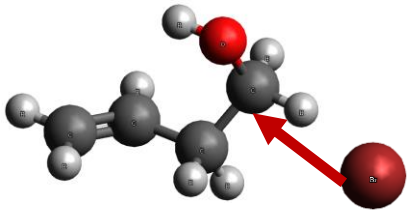
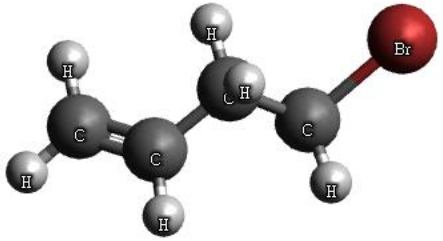
In the atom-based or fragment-based modifications, the molecular modification operators and pre-defined molecular building blocks (i.e. fragments and atoms) are repeatedly utilized to create new chemical structures in pursuit of property specifications. During this process, the changes of subject chemical structures are typically not based on the knowledge of realistic chemical reactions. Molpher[106, 107], Spaceship[108], MoleculeEvoluator[109], GraphGA[110, 111], GraphMCTS[110, 111], EvoMol[112] are the generative algorithms employing atom-based modifications, while CReM[105], LEADD[113], BRADSHAW[114], OpenGrowth[59], FOG[115], LigBuilder v3[116, 117], MOARF[118], PhDD[119], AutoGrow v3.0[120, 121], and Flux[60, 61] adopt fragment-based modifications. In contrast,

doi:10.6342/NTU202403528

reaction-based modifications are centered on templates derived from realistic chemical reactions.[122] Recent applications[123-127] of reaction-based modifications often make use of RDKit[64], where the reaction templates are encoded in "reaction SMARTS" format. When provided with a chemical structure along with a reaction template, RDKit identifies the substructures that align with the reactive site pattern defined in the template and execute the specified chemical transformation. DOGS[128, 129], AutoCouple[123], LiGen[130], and SYNOPSIS[131] are the exemplary models for reaction-based molecular design.

doi:10.6342/NTU202403528

Table 2.3-3. The three types of the generative algorithms.

Type	Starting chemical u_i	Operation δu_i	New chemical $u_i^{new} = u_i + \delta u_i$
Atom-based		e.g. substitution 	
Fragment-based	 amide linker  benzene	e.g. merge 	
Reaction-based		Example: a real reaction of nucleophilic substitution (**O**-[CH2;D2;+0:1]-[C:2]-[C:3]=[C:4])>>(**[Br;H0;D1;+0]**-[CH2;D2;+0:1]-[C:2]-[C:3]=[C:4]) 	

doi:10.6342/NTU202403528

2.3.4. Reverse Algorithm, Part (II): Selection Algorithms

Based on the categories of task specifications and solution methods, a review for literatures of fragment-based molecular design is summarized in **Table A3**. It should be noted that the brute force (BF) method, as adopted by R. Gani and E. A. Brignole (section 1.3.2), may experience combinatorial explosion when an enormous number of building blocks are available. In this situation, solving the CAMD problem with limited computational resources will be impractical. To address this issue, meta-heuristic algorithms can be used in place of the brute force method. Although meta-heuristic algorithms may not guarantee global optimality for solutions, they have reasonable trade-off between computational costs and optimality of solutions.[132] These algorithms include the genetic algorithm (GA)[133-138], simulated annealing algorithm (SA)[139, 140], genetic-simulated annealing composite algorithm (GA-SA)[141-144], ant colony optimization algorithm (ACO)[145-147], tabu search algorithm (TS)[48, 148-150], Monte Carlo tree search (MCTS)[110, 151-154], branch-and-reduce algorithm (B&R)[54, 155, 156], outer approximation (OA)[155, 157-160], and brutal-force (BF) search[161].

GA, SA, and ACO are inspired by the principles from nature. The main idea of GA is to simulate a molecular world governed by Darwinian theory. Specifically, a molecular structure is analogous to a chromosome. "Genetic operators" mimic genetic variations: a mutation operator modifies substructures of a molecular structure, creating a new molecular species, while a crossover operator exchanges the substructures between two parent molecules, yielding two new molecular species. After evaluating properties of these new molecules, a "selection algorithm" filters out undesirable species based on their performance rankings. This iterative generation-and-selection process continues until a collection of optimal species is identified.

doi:10.6342/NTU202403528

SA is based on the statistical mechanical interpretation of the annealing process. In the canonical ensemble (NVT system), the probability distribution of states follows the Boltzmann distribution:

$$P_i = \frac{\exp\left(-\frac{E_i^*}{T}\right)}{\sum_{states\ j}\ \exp\left(-\frac{E_j^*}{T}\right)} \tag{2.3–1}$$

Here, $E_{\mathrm{i}}^* = E_i/k_B$ represents the characteristic temperature for energy state E_i, with k_B being the Boltzmann constant. At high temperatures, it is equally probable for a physical system to be at any available quantum state. In contrast, at low temperatures, only the lowest-energy state is probable. The two cases show that the annealing ($T_{high} \rightarrow T_{low}$) is a process that gradually distinguishes among different energy states. In the context of an optimization task, this feature offers a wide range of potential solutions in the early stages and ensures satisfactory convergence in the later stages. In algorithm implementation, the E_i^* and E_j^* should be replaced with the objective function, and a suitable decaying rate α for temperature parameter T is necessary. A rapid decay in the T may lead to premature optimization, leading to convergence at the local minima near the initial guesses.

The concept of ACO is rooted in the foraging behavior exhibited by ants. Initially, the ant colony embarks on a random exploration of the area surrounding their nest, searching for a food source. Upon successful discovery, an ant returns to the nest carrying a bite of food, while simultaneously laying down pheromone trails along its path. These pheromone trails serve as a communication channel, attracting other ants to follow the same route. Once back at the nest, the ant resumes the exploration cycle. The

doi:10.6342/NTU202403528

attractiveness of a particular path is directly influenced by the concentration of pheromones deposited on it. However, this pheromone concentration naturally diminishes over time. Shorter paths between the nest and the food source offer the advantage of quicker travel times, leading to a higher concentration of pheromones remaining on these paths due to the shorter travel time. Consequently, a positive feedback loop is established: more ants are drawn to the shorter paths with higher pheromone concentrations, ultimately enabling the colony to identify the optimal route.

Unlike the previously discussed nature-inspired algorithms, Tabu Search (TS) employs a memory-based approach to navigate the search space for optimization problems. The central principle of TS revolves around the creation and utilization of a "tabu list." This list serves as a dynamic record of unfavorable moves or solutions encountered during the search process. By incorporating these elements into the taboo list, the algorithm prioritizes exploration of uncharted territories within the search space.

doi:10.6342/NTU202403528

Chapter 3. Constructing a Program for Conventional CAMD

In section 2.3, the diverse variety of each component in CAMD framework has been reviewed. We now construct the four components based on our own specifications.

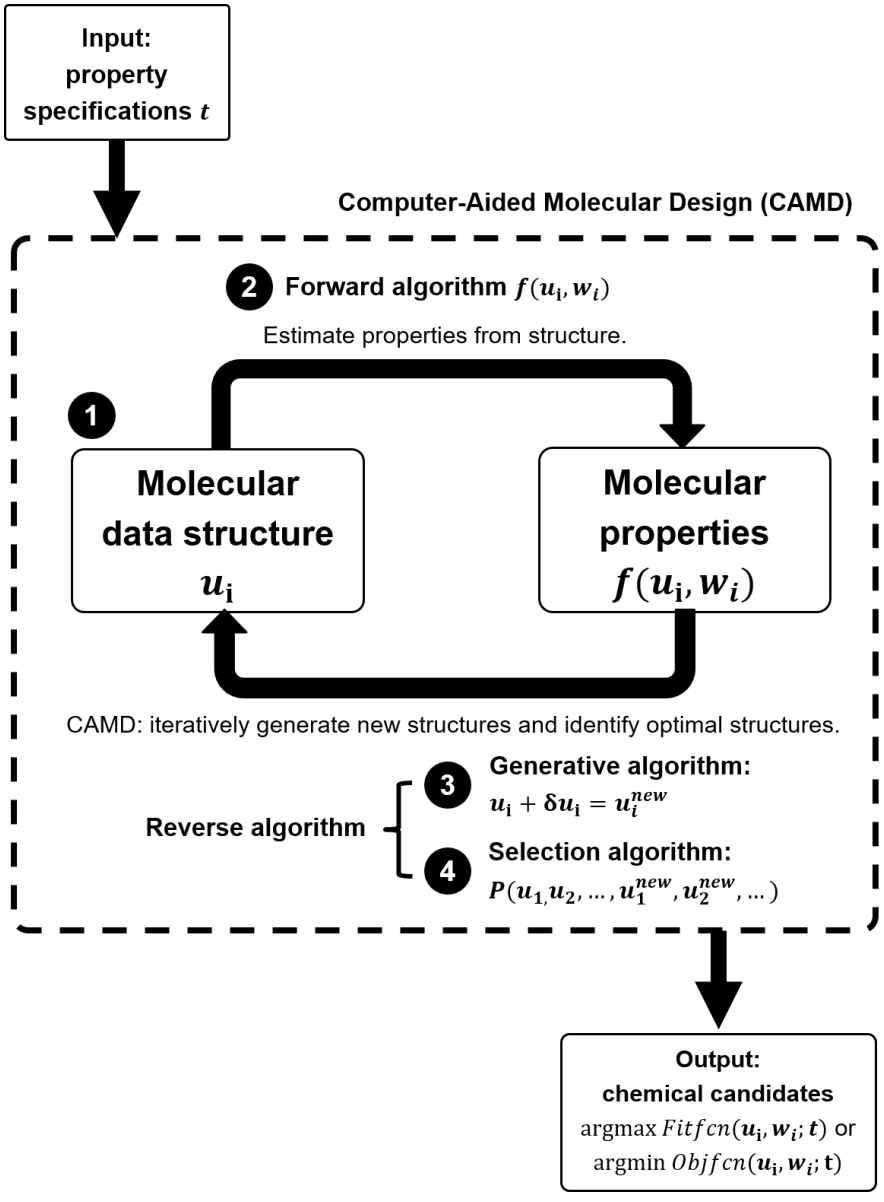


Figure 2.3-1. The four components in rule-based CAMD framework.

3.1. Chemical Representation: MARS+ Package

MARS (Molecular Assembling and Representation Suite)[162] serves as a versatile

doi:10.6342/NTU202403528

toolbox for general-purpose molecular design. It demonstrates the feasibility in using five arrays of integers to record the constituent elements along with their connectivity among the within a molecule, thereby forming a digital representation of molecular structure. Additionally, MARS provides a collection of operations that enable the permutation of molecular substructures, resulting in the creation of novel molecular species. One can experience this generative algorithm by inputting two (or more) chemical species and subjecting them to these structure operations.

This work presents MARS+[163], an extension of the original MARS software specifically designed to handle complex molecular structures. MARS+ expands its capabilities to encompass geometric isomers (cis-trans isomers), stereoisomers, complicated polycyclic compounds, and ionic species. This extended coverage of the chemical space allows MARS+ to explore a wider range of molecules with greater chemical and physical diversity.

At the core of MARS+ lies the molecular data structure (MDS), detailed in section 3.1.2. This MDS is comprised of eight integer arrays and two string arrays, efficiently storing information about atoms and fragments within a molecule. A key strength of MARS+ is its ability to combine multiple single-component MDS objects into a supermolecular MDS. This capability proves particularly valuable for handling complex chemicals, such as ionic liquids, which consist of separate cationic and anionic components. For the construction and manipulation of chemical structures, MARS+ offers a rich set of twelve operations. These operations can be categorized into three primary groups, as described in section 3.3:

- **9 uni-molecular operations:** These operations focus on modifying individual molecules through *addition*, *deletion*, or *insertion* of atoms, *bond change*s, *element*

doi:10.6342/NTU202403528

*substitution*s, *cis-trans inversion*, *chirality inversion*, *cyclization*, and *decyclization*.

- **2 bi-molecular operations:** These operations facilitate the redistribution of molecular fragments between two molecules, enabling the creation of new molecules through *crossover* and *combination*.

- **1 bi-supermolecular operation**: This operation, *component swap*, allows for the exchange of components between supermolecular entities.

To streamline molecule input and ensure the generation of canonical SMILES strings, MARS+ incorporates wrappers around selected Open Babel API functions. **Figure 3.1-1** provides a visual representation of the MARS+ architecture and its functionalities.

doi:10.6342/NTU202403528

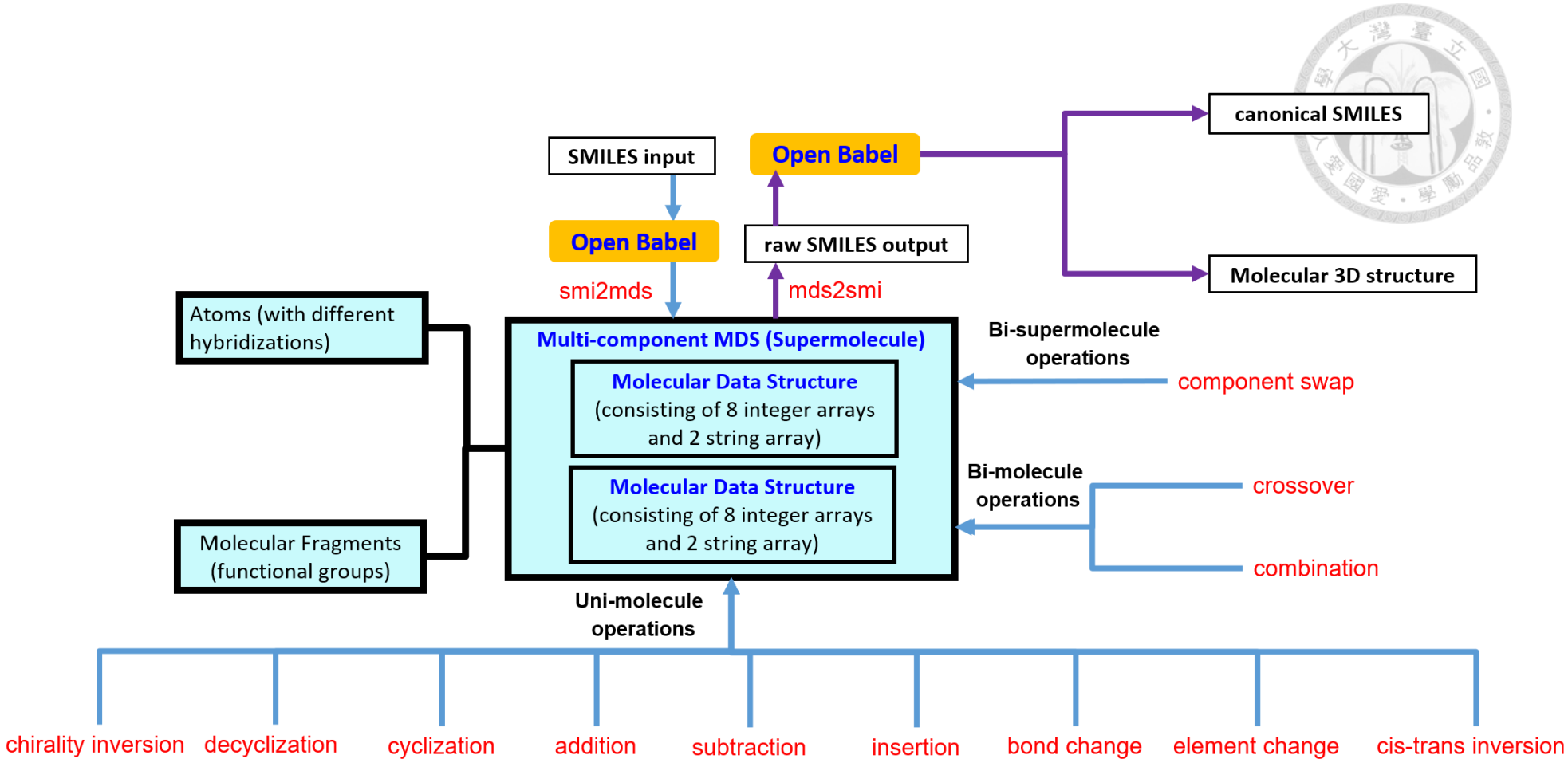


Figure 3.1-1. The architecture of MARS+ package. Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.

doi:10.6342/NTU202403528

3.1.1. The Library of Base Elements

MARS+ utilizes a library of pre-defined base elements to represent the building blocks of chemical structures. These base elements can encompass individual atoms (e.g., H, C, O), functional groups (e.g., CH3, OH, benzene), or even ionic groups (e.g., *1,3-dimethylimidazolium*). The definition of these base elements is established within the *set_up()* function in *src/ELEMENTS.cpp* source file. **Table A4** to **Table A6** in Appendix A provides a comprehensive list of neutral, cationic, and anionic base elements for reference. Each base element is characterized by a minimum of eight attributes, as summarized in **Table 3.1-1**.

Table 3.1-1. The attributes of a base element.

Attribute	Significance
name	The SMILES representation of the base element
id	A unique numerical identifier for the base element within the library
norder	The total number of valence bonds
order	An array storing the specific bond order for each valence bond
bd	An array storing number of valence bonds per order
index	The character index of the first bond in the *name* string.
suffspos	The starting position of any optional suffix
chg	The overall charge of the segment

It's important to note that not all attributes are utilized for every purpose. Here's a breakdown of the attributes crucial for different functionalities:

doi:10.6342/NTU202403528

- **MDS representation (section 3.1) and structural operations (section 3.3.1 to 3.3.3):** These functionalities primarily rely on the *id*, *norder*, *chg*, *order* array, and *bd* array for accurate representation and manipulation of the molecular data structure.

- **Raw SMILES output from MDS (section 3.3.4):** Generating raw SMILES strings from the MDS necessitates the *name*, *index*, *order*, and *suffspos* (presumably a total bond count) attributes.

The nomenclature of *name* string should follow the format of "[core_atom][valences][suffix]", where [core_atom] represents the central atom, [valences] indicates the available bond count, and [suffix] an optional component for specifying functional groups. This suffix allows for the flexible introduction of diverse functional groups into the base element library. **Figure 3.1-2** provides an illustrative example of the attribute settings for the base element *1,3-dimethylimidazolium* (*ID=36*).

The core concept behind MARS+ lies in representing each molecular input as a connectivity network constructed from these pre-defined base elements. Consequently, the comprehensiveness of the base element library directly influences the robustness and versatility of the MDS representation. Expanding the library with new elements is possible by following the instructions provided in the *inputs/element_lists/element_list*.txt file.

doi:10.6342/NTU202403528

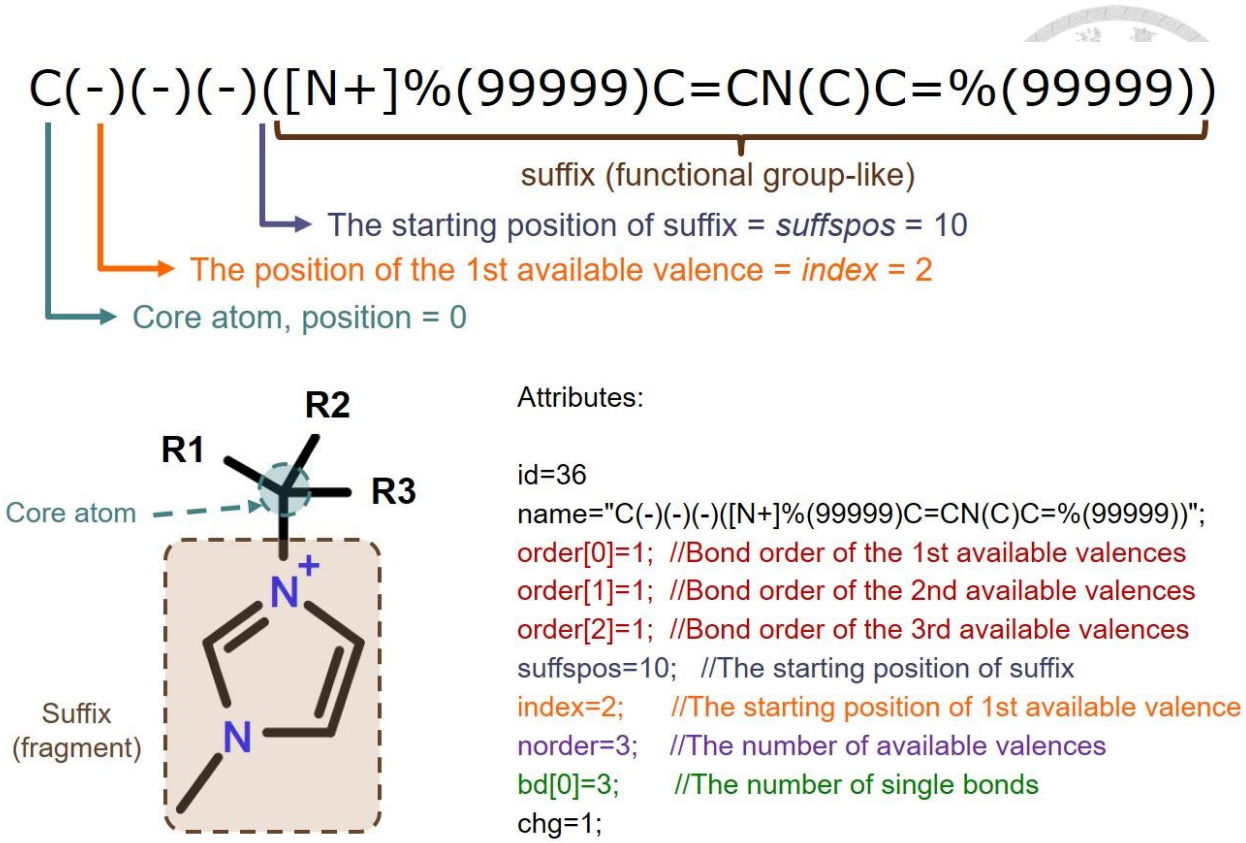


Figure 3.1-2. The attribute settings for the base element *1,3-dimethylimidazolium* (*id*=36). The last carbon in suffix uses a double bond to connect with [N+], forming a ring with 99999 as its default ring number. Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.

3.1.2. Molecular Data Structure (MDS)

In the MARS+ framework, the Molecular Data Structure (MDS) captures the connectivity between the fundamental elements of a molecule, akin to a molecular graph representation[72]. The MDS for single molecule is defined in *src/MOLECULE.h* source code file while the specialized supermolecule MDS, as exemplified by ionic liquids, is defined in *src/IL.h* source code file. The MDS consists of 10 data elements (see **Table 3.1-2**), each a one-dimensional array of size N. Here, N represents the total number of base elements forming the molecule. There's one exception: the ***cyclic bond order array***.

doi:10.6342/NTU202403528

The size of this particular array is dictated by the number of rings (N_r) present in the molecule, rather than the total number of base elements. In other words, the structure variable $\boldsymbol{u_i}$ is represented by,

$$\boldsymbol{u_i} = \begin{bmatrix} \boldsymbol{C_{1\times N}} \\ \boldsymbol{P_{1\times N}} \\ \boldsymbol{M_{1\times N}} \\ \boldsymbol{R_{1\times N}} \\ \boldsymbol{Chi_{1\times N}} \\ \boldsymbol{Cy_{1\times N}} \\ \boldsymbol{Cyb_{1\times N_r}} \\ \boldsymbol{Pr_{1\times N}} \\ \boldsymbol{Fct_{1\times N}} \\ \boldsymbol{Ect_{1\times N}} \end{bmatrix}_i \tag{3.1–1}$$

Table 3.1-2 provides a detailed explanation of each data element within the MDS. Each constituent element within a molecule is designated a unique integer ranging from 1 to N in the in the ***element indice array*** $\underline{\boldsymbol{C}}_{\boldsymbol{1\times N}}$ ***(variable: Cindex)***. The ***element type array*** $\underline{\boldsymbol{M}}_{\boldsymbol{1\times N}}$ ***(variable: Mindex)*** serves the purpose of recording the base element ID for each corresponding element within the ***element indice array***. This ID allows for the retrieval of information such as charge, name, and valence of each element from the base element library (see **Table A4** to **Table A6** in Appendix A). The ***parent indices array*** $\underline{\boldsymbol{P}}_{\boldsymbol{1\times N}}$ ***(variable: Pindex)*** stores the element index of the parent element within the molecule to which the current element is connected. It is important to note that each element within a molecule has exactly only one parent element, except for the first element in MDS (it has no parent element). In contrast, an element can be connected to two or more "descendant elements".

The ***bond order array*** $\underline{\boldsymbol{R}}_{\boldsymbol{1\times N}}$ ***(variable: Rindex)*** stores the bond order between an element and its parent element. The ***cyclic flag array*** $\underline{\boldsymbol{Cy}}_{\boldsymbol{1\times N}}$ ***(variable: Cyindex)***

doi:10.6342/NTU202403528

indicates cyclic substructures within the molecule. Each unique cyclic substructure is assigned a non-zero number to two of its member elements, indicating that the two elements are connected by a bond to form the ring. The bond order for this specific ring-forming bond is stored in the separate ***cyclic bond order array*** $\underline{\boldsymbol{Cyb}}_{\mathbf{1}\times N_r}$ ***(variable: Cybnd)***, where N_r represents the number of rings in the molecule. The ***cis-trans front/end flags array*** $\underline{\boldsymbol{Fct}}_{\mathbf{1}\times N}/\underline{\boldsymbol{Ect}}_{\mathbf{1}\times N}$ ***(variable: ctsisomer)*** specifies the "\" and "/" notation in front of or at the end of an element *name* to denote cis-trans isomerism. The ***protection flag array*** $\underline{\boldsymbol{Pr}}_{\mathbf{1}\times N}$ ***(variable: protect)*** identifies elements that are protected from any structural modifications during subsequent manipulations.

The OpenSMILES specification[74] incorporates the concept of "winding type" to represent the chirality of centers within a molecule. For a chiral carbon atom, "R1[C@](R2)(R3)(R4)" indicates that substituents R2, R3, and R4 are arranged in an anti-clockwise order when viewed from R1 towards the chiral carbon. Conversely, "R1[C@@](R2)(R3)(R4)" signifies a clockwise arrangement. This anti-clockwise winding ("@") is encoded as a value of 1, while clockwise winding ("@@") is encoded as 2 within the chirality flag array, ***chirality flag*** $\underline{\boldsymbol{Chi}}_{\mathbf{1}\times N}$ ***(variable: chi)***.

As an example, **Table 3.1-2** provides an illustrative example of the MDS representation for an imidazolium cation, C[n+](c1)ccn1[C@H](F)/C=C/C. Here's a breakdown of the information encoded within the data structure:

- **Element Indexing and Parentage**: Each atom in the cation is assigned a unique serial number (1 to 11) representing its element index. The parent index for the first element is always 0, signifying its position as the starting point in the data structure.

doi:10.6342/NTU202403528

- **Element Type and Base Element Library**: The element type array and the base element library (see **Table A4** to **Table A6** in Appendix A) work together to define the properties of each element. For instance, the first entry in the element type array, along with the base element library, identifies the first element (element index = 1) as "C(-)(-)(-)(-)" (ID = 1). This notation indicates a charge-neutral carbon atom with four single bonds. Similarly, the second element (element index = 2) is identified as "[N+](=)(-)(-)" (ID = 16), representing a positively charged nitrogen atom with a double bond and two single bonds. The parent element of the second element is the first element (element index=1), as indicated by the second entry of parent indices array.

- **Bond Orders**: The bond order array stores the bond order between an element and its parent. The second entry in this array indicates a single bond (value of 1) between the second element (element index = 2, [N+](=)(-)(-), ID = 16) and its parent (element index = 1, C(-)(-)(-)(-), ID = 1). Note that the bond order index for the first element is always zero, reflecting its status as the starting point.

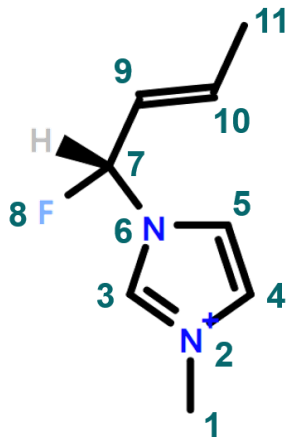
- **Cyclic Substructures**: The cyclic flag array and the cyclic bond order array work in tandem to represent cyclic structures within the molecule. In this example, the sixth element (element index = 6, N(-)(-)(-), ID = 7) and the third element (element index = 3, C(=)(-)(-), ID = 2) form a single bond to create the imidazolium ring. Consequently, both elements are assigned a value of 1 in the corresponding entry of the cyclic flag array. The bond order for this ring closure is stored as a single bond (value of 1) in the first entry of the cyclic bond order array.

doi:10.6342/NTU202403528

● **Protection Flags**: For the design of other imidazolium ionic liquids, the protection flag array can be employed. Assigning a value of 1 to all members of the imidazolium ring within this array protects them from modifications during operations like crossover, decyclization, or subtraction.

doi:10.6342/NTU202403528

Table 3.1-2. The molecular data structure (MDS) in MARS+ package

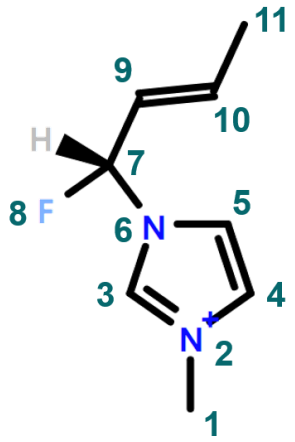
Name	Stored information	Canonical SMILES: C[n+](c1)ccn1[C@H](F)/C=C/C Molecular size: $N = 11$	
Element indices $\underline{\boldsymbol{C}}_{1\times N}$	Serial numbering from 1 to N for each base element in the molecule	1 2 3 4 5 6 7 8 9 10 11	 The numbers are the element indices
Parent indices $\underline{\boldsymbol{P}}_{1\times N}$	The base element to connect with	0 1 2 2 4 5 6 7 7 9 10	
Element types $\underline{\boldsymbol{M}}_{1\times N}$	The ID of the base elements (**Table A4** to **Table A6**)	1 16 2 2 2 7 1 11 2 2 1	
Bond orders $\underline{\boldsymbol{R}}_{1\times N}$	Bond order information for the connection determined by $(\underline{\boldsymbol{C}}_{1\times N}, \underline{\boldsymbol{P}}_{1\times N}, \underline{\boldsymbol{M}}_{1\times N})$	0 1 2 1 2 1 1 1 1 2 1	
Chirality flags $\underline{\boldsymbol{Chi}}_{1\times N}$	Chirality information of each base elements in the molecule	0 0 0 0 0 0 1 0 0 0 0	

†In this example the imidazolium ring is protected.

††The notation "_" signifies a null string rather than a blank space.

doi:10.6342/NTU202403528

Table 3.1-2. The molecular data structure (MDS) in MARS+ package (continued)

Name	Stored information	Canonical SMILES: C[n+](c1)ccn1[C@H](F)/C=C/C Molecular size: $N = 11$	
Cyclic flags $\underline{\boldsymbol{Cy}}_{1\times N}$	Ring numbering: Two elements labeled with the same number will be connected to form a ring.	0 0 1 0 0 1 0 0 0 0 0	 The numbers are the element indices
Cyclic bond orders $\underline{\boldsymbol{Cyb}}_{1\times N_r}$	The bond order for each cyclic flag	1	
Protection flags† $\underline{\boldsymbol{Pr}}_{1\times N}$	The elements labeled 1 will be free from genetic operations	0 1 1 1 1 1 1 0 0 0 0	
Cis-trans front flags†† $\underline{\boldsymbol{Fct}}_{1\times N}$	Record "/" or "\" that should be put in front of the element name	_ _ _ _ _ _ _ _ / _ _	
Cis-trans end flags†† $\underline{\boldsymbol{Ect}}_{1\times N}$	Record "/" or "\" that should be put at the end of element name (before the first bond)	_ _ _ _ _ _ _ _ _ / _	

†In this example the imidazolium ring is protected.

†† The notation "_" signifies a null string rather than a blank space.

doi:10.6342/NTU202403528

3.2. Forward Algorithm: Property Prediction Models

In this section, we present theoretical foundations for several crucial property models implemented in this work. All the models incorporated in this work are outlined in **Table 3.2-1** and **Table 3.2-2**.

Table 3.2-1. The property estimation method incorporated in this work

Property	Computational toolkits
Activity coefficient, $\gamma_{i/S}$	Gaussian solvation calculations[164] + COSMO-SAC model[100]
Highest occupied molecular orbital (HOMO), E_{HOMO}	Gaussian[164]
Lowest unoccupied molecular orbital (LUMO), E_{LUMO}	Gaussian[164]
Ionization potential (IP)[2], E_{IP}	Gaussian[164]
Electron affinity (EA)[2], E_{EA}	Gaussian[164]
Fundamental gap[2, 165], E_{gap}	Derived from IP and EA, $E_{gap} = E_{IP} - E_{EA}$
Electronegativity[97, 166], χ_m	Derived from IP and EA, $\chi_m = -\frac{E_{IP}+E_{EA}}{2}$
Chemical hardness[97, 167], η	Derived from IP and EA, $\eta = \frac{E_{IP}-E_{EA}}{2}$
Electrophilicity[97, 167], ω	Derived from IP and EA, $\omega = \frac{\chi_m^2}{2\eta} = \frac{(E_{IP}+E_{EA})^2}{4(E_{IP}-E_{EA})}$
SAscore[64, 168]	RDKit[64]
SCscore[169]	RDKit[64]
NPscore[64, 170]	RDKit[64]

doi:10.6342/NTU202403528

Table 3.2-2. The RDKit descriptors[171] incorporated in this work.

Type	Descriptors
Perception of substructure feature	Molecular Connectivity Chi Indexes, χ_v and χ_n Tanimoto Similarity, $\mathcal{T}_s(mol_1, mol_2)$ The number of atoms, N_{Atms} The number of rotatable bonds N_{RotBnd} The number of rings N_{Rings} The number of aromatic rings $N_{AroRings}$ The number of spiro atoms $N_{SpiAtms}$ The number of bridge atoms $N_{BriAtms}$, and so on
Quantification of structure asymmetry	Averaged distance from plane of best fit, PBF Principal moments of inertia, PMI_1, PMI_2, PMI_3 Inertial shape factor, ISF Eccentricity, $Eccent$ Asphericity, $Aspher$ Spherocity Index, $Sphero$
Estimation of physicochemical properties	1-octanol/water partition coefficient, $logP$ The number of hydrogen bond donors, N_{HBD} The number of hydrogen bond acceptors, N_{HBA} molecular refractive index, MR Labute's approximate surface area, ASA Topological polar surface area, $TPSA$ Radius of gyration, R_g

doi:10.6342/NTU202403528

3.2.1. COSMO-SAC Activity Coefficient Model

COSMO-SAC-2010 model[100, 172] serves as a reliable method for predicting activity coefficients based on the principles of solvation thermodynamics[173, 174]. This model requires only the chemical structure of the solute molecule as input. To understand the connection between the solvation process and activity coefficients, let's establish some key definitions. Firstly, the solvation is defined as an isothermal-isobaric process to transfer the solute molecules from a fixed position in an ideal gas phase into a fixed position within the solvent phase, without altering the solvent's composition.[174] The partial molar Gibbs free energy of a fixed solute molecule *i* in solvent *S* is denoted as $\overline{G}^*_{i/S}$. This term represents the pseudo-chemical potential of the solute, essentially the chemical potential $\overline{G}_{i/S}$ excluding the contribution from molecular translational motion (i.e. liberation free energy $RT\ln(x_i C_S \Lambda_i^3)$).

$$\overline{G}_{i/S}(T,P,\underline{x}) = \overline{G}^*_{i/S}(T,P,\underline{x}) + RT\ln(x_i C_S \Lambda_i^3) \tag{3.2–1}$$

Here, x_i is the mole fraction of species *i* in the solvent *S*, C_S is the number density of solvent molecules, $\Lambda_i = \frac{\hbar}{\sqrt{2\pi m_i kT}}$ denotes the thermal wavelength of solute *i*, $\hbar$ represents the reduced Plank constant, m_i represent the mass of a solute particle, and k denotes the Boltzmann constant. The pseudo-chemical potential $\overline{G}^*_{i/S}$ is the free energy associated with intramolecular degrees of freedom (e.g. vibrational, rotational, electronic, and nuclear contributions) and intermolecular interactions. On the other hand, the liberation free energy $RT\ln(x_i C_S \Lambda_i^3)$ gets its name as it can be interpreted as the required

doi:10.6342/NTU202403528

work to enable thermal translational motion of solute in solvent phase. With these definitions, we can now define the solvation free energy $\Delta\overline{G}_{i/S}^{*,sol}$. It represents the difference between the pseudo-chemical potential of the solute in the solution state $\overline{G}_{i/S}^{*}(T,P,\underline{x})$ and the ideal gas state $\overline{G}_{i/S}^{*,IGM}(T,P,\underline{x})$.

$$\Delta\overline{G}_{i/S}^{*,sol}(T,P,\underline{x}) = \overline{G}_{i/S}^{*}(T,P,\underline{x}) - \overline{G}_{i/S}^{*,IGM}(T,P,\underline{x}) \tag{3.2–2}$$

Combining eq (5.2–1) and (5.2–2), we have:

$$\Delta\overline{G}_{i/S}^{*,sol}(T,P,\underline{x}) = \overline{G}_{i/S}(T,P,\underline{x}) - \overline{G}_{i/S}^{IGM}(T,P,\underline{x}) + RT\ln\left(\frac{P}{C_S kT}\right) \tag{3.2–3}$$

According to the definition of activity and eq (3.2–3):

$$\begin{aligned}\ln x_i\gamma_{i/S}(T,P,\underline{x}) &= \frac{\overline{G}_{i/S}(T,P,\underline{x}) - \underline{G}_{i/i}(T,P)}{RT} \\ &= \frac{\Delta\overline{G}_{i/S}^{*,sol}(T,P,\underline{x}) - \Delta\underline{G}_{i/i}^{*,sol}(T,P)}{RT} + \frac{\overline{G}_{i/S}^{IGM}(T,P,\underline{x}) - \underline{G}_{i/i}^{IG}(T,P)}{RT} + \ln\frac{C_S}{C_i}\end{aligned} \tag{3.2–4}$$

Here, $\underline{G}_{i/i}$ represents the molar free energy of pure fluid *i*. Given $\overline{G}_{i/S}^{IGM}(T,P,\underline{x}) - \underline{G}_{i/i}^{IG}(T,P) = RT\ln x_i$, the activity coefficient of the solute *i* in the solvent *S* is determined from the free energy difference between the solvated solution state $\Delta\overline{G}_{i/S}^{*sol}$ and the solvated pure fluid state $\Delta\underline{G}_{i/i}^{*,sol}$.[173, 175]

doi:10.6342/NTU202403528

$$\ln \gamma_{i/S}(T,P,\underline{x}) = \frac{\Delta \underline{G}_{i/S}^{*sol}(T,P,\underline{x}) - \Delta \underline{G}_{j/i}^{*sol}(T,P)}{RT} + \ln \frac{C_S}{C_i} \tag{3.2–5}$$

The solvation of solute *i* into the implicit continuum solvent *S* can be decomposed into 7 steps based on COSMO solvation theory[101, 175, 176]: (a) the charge of the ideal-gas solute is turned off. (b) cavity is created within the solvent *S* in order to accommodate the solute molecule, resulting in the cavity formation free energy $\Delta \overline{G}_{i/S}^{*cav}$. (c) the charge-neutral solute is transferred from the gas phase to the cavity in the solvent phase. (d) The solvent *S* is transformed into a perfect conductor with dielectric constant of infinity. (e) the charge of the solute is turned on and is completely screened by the perfect conductor, resulting in the free energy of ideal solvation $\Delta \overline{G}_i^{*is} = E_i^{COSMO} - E_i^{IG}$. (f) the charge density σ_n^* on each ideal screening surface segment n is averaged using eq. (3.2–6), resulting in the apparent charge density σ_n and charge averaging correction term $\Delta \overline{G}_i^{*cc}$. (g) the averaged screening charges are removed to restore the original solvent *S*, resulting in the restoring free energy $\Delta \overline{G}_{i/S}^{*res}$.

$$\sigma_n = \frac{\sum_{\sigma_m^*} \sigma_m^* \frac{r_m^2 r_{eff}^2}{r_m^2 + r_{eff}^2} \exp\left(-f_{decay} \frac{d_{nm}}{r_m^2 + r_{eff}^2}\right)}{\sum_{\sigma_m^*} \frac{r_m^2 r_{eff}^2}{r_m^2 + r_{eff}^2} \exp\left(-f_{decay} \frac{d_{nm}}{r_m^2 + r_{eff}^2}\right)} \tag{3.2–6}$$

Here, d_{nm} is the distance (in Å) between the ideal screening surface segments n and m. $r_{eff} = \left(a_{eff}/\pi\right)^{0.5}$ is the effective radius of each surface segment, where $a_{eff} = 7.25$ Å^2 is the effective surface area of each segment. The unit conversion facto $f_{decay} = 3.57$ corrects the distance d_{mn} from Å to Bohr radius.[172] If dispersive

doi:10.6342/NTU202403528

interactions $\Delta\overline{G}_{i/S}^{*dsp}$ are also considered[177-179], the overall solvation free energy can be expressed as the sum of the five contributions.

$$\Delta\overline{G}_{i/S}^{*sol} = \Delta\overline{G}_{i}^{*is} + \Delta\overline{G}_{i/S}^{*res} + \Delta\overline{G}_{i}^{*cc} + \Delta\overline{G}_{i/S}^{*dsp} + \Delta\overline{G}_{i/S}^{*cav} \quad (3.2\text{–}7)$$

In particular, the sum of the first four terms is referred to as the solvation charging free energy $\Delta\overline{G}_{i/S}^{*chg} = \Delta\overline{G}_{i}^{*is} + \Delta\overline{G}_{i/S}^{*res} + \Delta\overline{G}_{i}^{*cc} + \Delta\overline{G}_{i/S}^{*dsp}$, as it originates from (either permanent or transient) charges and dipoles. On the other hand, the cavity formation energy $\Delta\overline{G}_{i/S}^{*cav}$ accounts for the molecular size and shape differences among components. Since the free energy of ideal solvation $\Delta\overline{G}_{i}^{*is}$ and the charge averaging free energy $\Delta\overline{G}_{i}^{*cc}$ are only dependent on solute species, they will be cancelled out in the calculation of eq. (3.2–5). The dispersion term $\Delta\overline{G}_{i/S}^{*dsp}$ is assumed to be a weak function of solvent (i.e. $\Delta\overline{G}_{i/S}^{*dsp} \approx \Delta\underline{G}_{i/i}^{*,dsp}$) in COSMO-SAC-2010 model, therefore it also assumed to be cancelled out in eq. (3.2–5). Nevertheless, the contribution of dispersion term to the activity coefficient is explicitly considered in the later development.[179] From these arguments, eq. (3.2–5) can be rewritten as eq. (3.2–8).

$$\ln\gamma_{i/S} = \frac{\Delta\underline{G}_{i/S}^{*res} - \Delta\underline{G}_{i/i}^{*res}}{RT} + \frac{\Delta\underline{G}_{i/S}^{*cav} - \Delta\underline{G}_{i/i}^{*cav}}{RT} + \ln\frac{C_S}{C_i} \quad (3.2\text{–}8)$$

In particular, Lin and Sandler[175, 180] suggest use Staverman-Guggenheim model[181, 182] to describe the cavity formation term along with the concentration term in eq. (3.2–8).

doi:10.6342/NTU202403528

$$\ln \gamma_{i/S}{}^{SG} = \frac{\Delta \underline{G}_{j/S}^{*cav} - \Delta \underline{G}_{j/i}^{*cav}}{RT} + \ln \frac{C_S}{C_i}$$
$$= \ln\left(\frac{\Phi_i}{x_i}\right) + \frac{z}{2} q_i \ln\left(\frac{\theta_i}{\phi_i}\right) + l_i - \frac{\Phi_i}{x_i} \sum_j x_j l_j \quad (3.2\text{–}9)$$

Here, $\theta_i = x_i q_i / \sum x_i q_i$, $\phi_i = x_i r_i / \sum x_i r_i$, $l_i = (z/2)(r_i - q_i) - (r_i - 1)$, $r_i = V_i^{COSMO}/r_0$, $q_i = A_i^{COSMO}/q_0$. x_i is the mole fraction of species *i*, r_i is the normalized volume of species *i*, $r_0 = 66.69$ Å^3 is the reference volume, q_i is the surface area parameters for *i*, $q_0 = 79.53$ Å^2 is the reference area, and $z = 10$ is the parameter of coordination number. V_i^{COSMO} and A_i^{COSMO} are the volume and the surface area of species *i* obtained from COSMO calculations, respectively.

On the other hand, the determination of restoring free energy $\Delta \overline{G}_{i/S}^{*res}$ is based on the screening charge surface obtained from COSMO solvation calculation. The total surface area of a solute molecule *i*, denoted as A_i, can be factored into three contributions[100]: $A_i^{NHB}(\sigma)$ from surface not involved in hydrogen bonding, $A_i^{OH}(\sigma)$ from surface involved in OH-typed hydrogen-bonding, and $A_i^{OT}(\sigma)$ from surface involved in HF-typed, NH-typed, as well as other special hydrogen bonding such as O in ketones and NO_2 in nitro compounds. The σ-profile $p_i(\sigma)$, i.e. the probability distribution of finding a surface with charge density σ, can be obtained after a reweighting by a gaussian function $p^{hb}(\sigma)$,

$$P^{hb}(\sigma) = 1 - \exp\left(\frac{\sigma^2}{2\sigma_o^2}\right) \quad (3.2\text{–}10)$$

doi:10.6342/NTU202403528

$$p_i^{NHB}(\sigma) = \frac{A_i^{nhb}(\sigma) + A_i^{hb}(\sigma)[1 - P^{hb}(\sigma)]}{A_i} \tag{3.2–11}$$

$$p_i^{OH}(\sigma) = \frac{A_i^{OH}(\sigma)P^{hb}(\sigma)}{A_i} \tag{3.2–12}$$

$$p_i^{OT}(\sigma) = \frac{A_i^{OT}(\sigma)P^{hb}(\sigma)}{A_i} \tag{3.2–13}$$

$$p_i(\sigma) = p_i^{NHB}(\sigma) + p_i^{OH}(\sigma) + p_i^{OT}(\sigma) \tag{3.2–14}$$

Here, $\sigma_0 = 0.007\ e/\text{Å}$. For a mixture system, its σ-profile is superposed by the surface area distribution of each pure component *i*, with their mole fraction x_i as the weighting factor.

$$p_S(\sigma) = \frac{\sum_i^c x_i A_i p_i(\sigma)}{\sum_i^c x_i A_i} \tag{3.2–15}$$

The σ-profile is utilized to calculate the electrostatic interactions between segments σ_m^t and σ_n^k. Here, the subscript *m* (or *n*) refers to the particular surface segment *m* (or *n*), and the superscript *t* (or *k*) indicates the type of the segment, i.e. *OH*, *OT*, or *NHB*. It is important to notice that, in COSMO-SAC model, the chemical species in the system are regarded as a mixture of the charged surface segments. The probability for two surface segments to form an interacting pair is modeled by Boltzmann distribution, as described by eq. (3.2–16). Each possible segment pair is microstate assumed to be independent from the other pairs.

$$p_S(\sigma_m^t)p_S(\sigma_n^k) = \exp\left[-\frac{E_{pair}(\sigma_m^t, \sigma_n^k) - \left(\mu_s(\sigma_m^t) + \mu_s(\sigma_n^t)\right)}{kT}\right] \tag{3.2–16}$$

doi:10.6342/NTU202403528

Here, $E_{pair}(\sigma_m^t, \sigma_n^k)$ is the self-energy of segment pair (σ_m^t, σ_n^k), $\mu_s(\sigma_m^t)$ represents the segment chemical potential of fragment σ_m^t. Self-energy can be factorized into the misfit energy $E_{mf}(\sigma_m^t, \sigma_n^k, T)$, hydrogen bonding interaction $E_{hb}(\sigma_m^t, \sigma_n^k)$, and non-electrostatic energy (mostly dispersion) E_{ne}.

$$\begin{aligned} E_{pair}(\sigma_m^t, \sigma_n^k) &= E_{mf}(\sigma_m^t, \sigma_n^k, T) + E_{hb}(\sigma_m^t, \sigma_n^k) + E_{ne} \\ &= \left(A_{ES} + \frac{B_{ES}}{T^2}\right)(\sigma_m^t + \sigma_n^k)^2 - c_{hb}(\sigma_m^t, \sigma_n^k)(\sigma_m^t - \sigma_n^k)^2 + E_{ne} \end{aligned} \tag{3.2–17}$$

To define the segment activity coefficient $\Gamma_S(\sigma_m^t)$ for segment σ_m^t, the charge-neutral ideal segment mixture (CNISM), where the partial molar free energy of fragment σ_m^t is $\mu_s^{CNISM}(0) = \mu_s^0(0) + kT\ln p_s(\sigma_m^t)$, is chosen as the reference system, as shown in eq. (3.2–18). $\mu_s^0(0) = \frac{1}{2}E_{pair}(0,0)$ is the chemical potential of a pure segment species under charge-neutral condition.

$$\ln\Gamma_S(\sigma_m^t) = \frac{\mu_s(\sigma_m^t) - \left(\mu_s^0(0) + kT\ln p_s(\sigma_m^t)\right)}{kT} \tag{3.2–18}$$

Note that p_S, $\mu_s^0(0)$, and E_{pair} in eq (3.2–16) and eq (3.2–18) are known information from COSMO calculation. Therefore, it is intuitive to solve eq (3.2–16) first for the chemical potential terms, and then substitute the results into eq (3.2–18) to calculate $\ln\Gamma_S$. However, this will lead to a large system of equations. To see this, let the number of surface segments with charge density σ_q be n_q, the total number of surface segments be $f = \sum_q n_q$, and $n_{qv} = n_{vq} = \frac{f}{2}p_S(\sigma_q)p_S(\sigma_v)$ be the number of pairs

doi:10.6342/NTU202403528

forming from segment q and v. f and n_q are also the known information obtained from COSMO calculation. The n_{qv} terms contribute $f(f+1)/2$ unknown variables, and the conservation law of segments provide f independent equations:

$$\begin{aligned} &n_{11} + n_{12} + \cdots + n_{1f} = n_1 \\ &n_{21} + n_{22} + \cdots + n_{2f} = n_2 \\ &\vdots \\ &n_{f1} + n_{f2} + \cdots + n_{ff} = n_f \end{aligned} \tag{3.2–19}$$

In addition, there are f unknown chemical potential terms $\mu_s(\sigma_q)$, and eq (3.2–16) provides $f(f+1)/2$ independent equations to relate them. It turns out that there are $f(f+3)/2$ equations and $f(f+3)/2$ unknowns are to be solved. The number of surface segments for a medium-sized molecule can be up to 3000, therefore the system of equations might be impractical to solve. An alternative mathematical form for $\Gamma_S(\sigma_m^t)$ is found to facilitate the calculation and enhance the robustness. The key idea is to eliminate the chemical potential terms $\mu_s(\sigma_m^t)$ and $\mu_s(\sigma_n^t)$ while combining eq (3.2–16) with (3.2–18). Firstly, summing over all the σ_n^k in eq (3.2–16) can eliminate $p_S(\sigma_n^k)$ because of the relationship $\sum_{\sigma_n^k} p_S(\sigma_n^k) = 1$. This leads to an expression for $\mu_s(\sigma_m^t)$:

$$\mu_s(\sigma_m^t) = kT\ln p_S(\sigma_m^t) - kT\ln\left[\sum_{\sigma_n^k} \exp\left(-\frac{E_{pair}(\sigma_m^t, \sigma_n^k) - \mu_s(\sigma_n^t)}{kT}\right)\right] \tag{3.2–20}$$

Substituting eq (3.2–20) for the $\mu_s(\sigma_m^t)$ in eq (3.2–18), we have:

doi:10.6342/NTU202403528

$$\ln\Gamma_S(\sigma_m^t) = -\ln\left[\sum_{\sigma_n^k} \exp\left(-\frac{E_{pair}(\sigma_m^t, \sigma_n^k) - \mu_s(\sigma_n^t) + \mu_s^0(0)}{kT}\right)\right] \quad (3.2\text{–}21)$$

Using eq (3.2–18) to eliminate $\mu_s(\sigma_n^t)$ term in (3.2–21), we finally arrive at eq. (3.2–22). This equation enables us to determine $\ln\Gamma_S(\sigma_m^t)$ by successive iterations. Specifically, $\Gamma_S(\sigma_n^k)$ is initialized with 0 for every σ_n^k, and then the formula on the right-hand side is continually used to update $\Gamma_S(\sigma_m^t)$.

$$\ln\Gamma_S(\sigma_m^t) = -\ln\left[\sum_{\sigma_n^k} p_S(\sigma_n^k)\Gamma_S(\sigma_n^k) \exp\left(-\frac{\Delta W(\sigma_m^t, \sigma_n^k)}{k_B T}\right)\right] \quad (3.2\text{–}22)$$

Here, $\Delta W(\sigma_m^t, \sigma_n^k) = E_{pair}(\sigma_m^t, \sigma_n^k) - E_{pair}(0,0)$ represents the exchange energy between fragments σ_m^t and σ_n^k.

$$\Delta W(\sigma_m^t, \sigma_n^k) = \left(A_{ES} + \frac{B_{ES}}{T^2}\right)(\sigma_m^t + \sigma_n^k)^2 - c_{hb}(\sigma_m^t, \sigma_n^k)(\sigma_m^t - \sigma_n^k)^2 \quad (3.2\text{–}23)$$

Once the successive iterations in eq. (3.2–22) reach convergence, the restoring energy $\Delta\underline{G}_{i/S}^{*res}$ can be expressed in terms of fragment contributions:

$$\frac{\Delta\underline{G}_{i/S}^{*res}}{RT} = \frac{A_i}{a_{eff}} \sum_{\sigma_m^t} p_i(\sigma_m^t)\ln\Gamma_S(\sigma_m^t) \quad (3.2\text{–}24)$$

Finally, combining eq (3.2–8), eq (3.2–9) , and eq (3.2–24), we have:

doi:10.6342/NTU202403528

$$\ln \gamma_{i/S} = \frac{A_i}{a_{eff}} \sum_{\sigma_m^t} p_i(\sigma_m^t)[\ln \Gamma_S(\sigma_m^t) - \ln \Gamma_i(\sigma_m^t)] + \ln \gamma_{i/S}{}^{SG} \quad (3.2\text{–}25)$$

The electrostatic interaction parameters, namely A_{ES}, B_{ES}, $c_{hb}(\sigma_m^t, \sigma_n^k)$, the value σ_0 in the Gaussian function for fragment classification, and the effective interaction area a_{eff} between the two fragments, are the few parameters required by the COSMO-SAC model and can be found in previous literature [100, 175].

3.2.2. Electronic Properties from Quantum Simulations

The electron density $\rho(\mathbf{r})$ is an important property in quantum mechanical calculations, as it determines complete information of a ground state (including the external field $v(\mathbf{r})$ arisen from presence of nuclei) according to Hohenberg-Kohn theorems.[183] Its mathematical form is expressed as eq (3.2–26).

$$\rho(\mathbf{r}) = N|\Psi(\mathbf{s_1}, \mathbf{r_1}, \dots, \mathbf{s_N}, \mathbf{r_N})|^2 \quad (3.2\text{–}26)$$

$$N = \int \rho(\mathbf{r}) d\boldsymbol{r} \quad (3.2\text{–}27)$$

Here, N is the number of elections in the considered system, Ψ is the normalized wave function for the system, $\mathbf{s_i}$ represents the spin state of electron *i*, $\mathbf{r_i}$ is the coordinate of electron *i*, $d\boldsymbol{r} = d\boldsymbol{s_1} d\boldsymbol{r_1} \cdots d\boldsymbol{s_N} d\boldsymbol{r_N}$. Based on density functional theory (DFT)[183], The total energy functional $E[\rho(\mathbf{r})]$ is expressed as eq (3.2–43).

doi:10.6342/NTU202403528

$$E[\rho(\mathbf{r})] = T[\rho(\mathbf{r})] + V_{ee}[\rho(\mathbf{r})] + V_{ne}[\rho(\mathbf{r})]$$
$$= T[\rho(\mathbf{r})] + V_{ee}[\rho(\mathbf{r})] + \int \rho(\mathbf{r}) v(\mathbf{r}) d\mathbf{r} \quad (3.2\text{–}28)$$

Here, $T[\rho(\mathbf{r})]$ is total kinetic energy of the electron system, $V_{ee}[\rho(\mathbf{r})]$ is the total electron-electron repulsive energy, and $v(\mathbf{r})$ is external field (in this case, electric field produced by the nuclei). With the functional form $v(\mathbf{r})$ fixed, we can use Lagrange multiplier method to find the lowest total energy $E[\rho(\mathbf{r})]$ subjected to eq (3.2–27).

$$\mathcal{L}[\rho(\mathbf{r}), \chi_m] = E[\rho(\mathbf{r})] - \chi_m \left(\int \rho(\mathbf{r}) d\mathbf{r} - N \right) \quad (3.2\text{–}29)$$

Here, $\mathcal{L}[\rho(\mathbf{r}), \chi_m]$ is the Lagrangian, χ_m is the Lagrange multiplier for condition eq (3.2–27). The functional derivate[184] of $\mathcal{L}[\rho(\mathbf{r}), \chi_m]$ with respect to $\rho(\mathbf{r})$ is:

$$\frac{\delta E[\rho(\mathbf{r})]}{\delta \rho(\mathbf{r})} - \int \chi_m d\mathbf{r} = 0 \quad (3.2\text{–}30)$$

On the other hand. we have from eq (3.2–28) that:

$$\frac{\delta E[\rho(\mathbf{r})]}{\delta v(\mathbf{r})} = \int \rho(\mathbf{r}) d\mathbf{r} \quad (3.2\text{–}31)$$

It turns out the total energy functional $E[\rho(\mathbf{r}), v(\mathbf{r})]$ can be recasted as[166]:

doi:10.6342/NTU202403528

$$
\begin{aligned}
E[\rho(\mathbf{r}), v(\mathbf{r})] &= \int_{\rho=0,v=0}^{\rho(\mathbf{r}),v(\mathbf{r})} \delta E[\rho(\mathbf{r}), v(\mathbf{r})] \\
&= \int_{\rho=0,v=0}^{\rho(\mathbf{r}),v(\mathbf{r})} \frac{\delta E[\rho(\mathbf{r}), v(\mathbf{r})]}{\delta \rho(\mathbf{r})} \delta\rho(\mathbf{r}) + \frac{\delta E[\rho(\mathbf{r}), v(\mathbf{r})]}{\delta v(\mathbf{r})} \delta v(\mathbf{r}) \\
&= \chi_m \int \int_{\rho=0}^{\rho(\mathbf{r})} \delta\rho(\mathbf{r})\, d\mathbf{r} + \int \rho(\mathbf{r}) \int_{v=0}^{v(\mathbf{r})} \delta v(\mathbf{r})\, d\mathbf{r} \\
&= \chi_m N + \int \rho(\mathbf{r}) v(\mathbf{r}) d\mathbf{r}
\end{aligned}
\tag{3.2–32}
$$

Here, $\chi_m = (\partial E/\partial N)_v$ is defined as electronegativity by R. S. Mulliken.[185] In analogy to thermodynamic depiction, χ_m bears the significance of "chemical potential" for the electron system. Now use Taylor expansion to obtain expression for the energy functional of anion state $E[\rho_{N+1}(\boldsymbol{r})]$ and cation state $E[\rho_{N-1}(\boldsymbol{r})]$. Let $\rho_{N+1}(\boldsymbol{r}) = \rho_N(\boldsymbol{r}) + \Delta\rho_+(\boldsymbol{r})$ and $\rho_{N-1}(\boldsymbol{r}) = \rho_N(\boldsymbol{r}) - \Delta\rho_-(\boldsymbol{r})$, then,

$$
\begin{aligned}
E[\rho_{N+1}(\boldsymbol{r})] &= E[\rho_N(\boldsymbol{r}) + \Delta\rho_+(\boldsymbol{r})] \\
&= E[\rho_N(\boldsymbol{r})] + \int \left(\frac{\delta E[\rho_N(\boldsymbol{r})]}{\delta \rho_N(\boldsymbol{r})}\right)_v \Delta\rho_+(\boldsymbol{r}) d\boldsymbol{s} d\mathbf{r} \\
&+ \frac{1}{2}\int \left(\frac{\delta^2 E[\rho_N(\boldsymbol{r})]}{\delta\rho_N(\boldsymbol{r}')\delta\rho_N(\boldsymbol{r}'')}\right)_v \Delta\rho_-(\boldsymbol{r}')\Delta\rho_-(\boldsymbol{r}'') d\boldsymbol{s}' d\boldsymbol{r}' d\boldsymbol{s}'' d\mathbf{r}'' + \cdots
\end{aligned}
\tag{3.2–33}
$$

$$
\begin{aligned}
E[\rho_{N-1}(\boldsymbol{r})] &= E[\rho_N(\boldsymbol{r}) - \Delta\rho_-(\boldsymbol{r})] \\
&= E[\rho_N(\boldsymbol{r})] - \int \left(\frac{\delta E[\rho_N(\boldsymbol{r})]}{\delta \rho_N(\boldsymbol{r})}\right)_v \Delta\rho_-(\boldsymbol{r}) d\boldsymbol{s} d\mathbf{r} \\
&+ \frac{1}{2}\int \left(\frac{\delta^2 E[\rho_N(\boldsymbol{r})]}{\delta\rho_N(\boldsymbol{r}')\delta\rho_N(\boldsymbol{r}'')}\right)_v \Delta\rho_-(\boldsymbol{r}')\Delta\rho_-(\boldsymbol{r}'') d\boldsymbol{s}' d\boldsymbol{r}' d\boldsymbol{s}'' d\mathbf{r}'' + \cdots
\end{aligned}
\tag{3.2–34}
$$

The mathematical forms for ionization potential ($E_{IP} = E\left[\rho_{N-1}(\boldsymbol{r})\right] - E\left[\rho_N(\boldsymbol{r})\right]$) and

doi:10.6342/NTU202403528

electron affinity ($E_{EA} = E[\rho_N(\boldsymbol{r})] - E[\rho_{N+1}(\boldsymbol{r})]$) can be determined based on eq (3.2–33) and (3.2–34), namely:

$$\begin{aligned} E_{EA} &= E[\rho_N(\boldsymbol{r})] - E[\rho_{N+1}(\boldsymbol{r})] \\ &= -\int \left(\frac{\delta E[\rho_N(\boldsymbol{r})]}{\delta \rho_N(\boldsymbol{r})}\right)_v \Delta\rho_+(\boldsymbol{r}) d\boldsymbol{s} d\mathbf{r} \\ &\quad -\frac{1}{2}\int \left(\frac{\delta^2 E[\rho_N(\boldsymbol{r})]}{\delta\rho_N(\boldsymbol{r}')\delta\rho_N(\boldsymbol{r}'')}\right)_v \Delta\rho_-(\boldsymbol{r}')\Delta\rho_-(\boldsymbol{r}'') d\boldsymbol{s}' d\boldsymbol{r}' d\boldsymbol{s}'' d\mathbf{r}'' + \cdots \end{aligned} \tag{3.2–35}$$

$$\begin{aligned} E_{IP} &= E[\rho_{N-1}(\boldsymbol{r})] - E[\rho_N(\boldsymbol{r})] \\ &= -\int \left(\frac{\delta E[\rho_N(\boldsymbol{r})]}{\delta \rho_N(\boldsymbol{r})}\right)_v \Delta\rho_-(\boldsymbol{r}) d\boldsymbol{s} d\mathbf{r} \\ &\quad +\frac{1}{2}\int \left(\frac{\delta^2 E[\rho_N(\boldsymbol{r})]}{\delta\rho_N(\boldsymbol{r}')\delta\rho_N(\boldsymbol{r}'')}\right)_v \Delta\rho_-(\boldsymbol{r}')\Delta\rho_-(\boldsymbol{r}'') d\boldsymbol{s}' d\boldsymbol{r}' d\boldsymbol{s}'' d\mathbf{r}'' + \cdots \end{aligned} \tag{3.2–36}$$

The electronegativity χ_m is determined as the arithmetic average of mean $\left(\frac{\delta E[\rho_N(\boldsymbol{r})]}{\delta \rho_N(\boldsymbol{r})}\right)_v$ in eq (3.2–35) and eq (3.2–36).

$$\chi_m = -\left(\frac{\partial E}{\partial N}\right)_v \approx \frac{1}{2}\left(-\frac{E_{EA}}{\int \Delta\rho_+(\boldsymbol{r}) d\boldsymbol{s} d\mathbf{r}} - \frac{E_{IP}}{\int \Delta\rho_-(\boldsymbol{r}) d\boldsymbol{s} d\mathbf{r}}\right) = -\frac{E_{IP} + E_{EA}}{2} \tag{3.2–37}$$

Note that $\int \Delta\rho_+(\boldsymbol{r}) d\boldsymbol{s} d\mathbf{r} = \int \Delta\rho_-(\boldsymbol{r}) d\boldsymbol{s} d\mathbf{r} = 1$. Similarly, the arithmetic average of mean $\left(\frac{\delta^2 E[\rho_N(\boldsymbol{r})]}{\delta\rho_N(\boldsymbol{r}')\delta\rho_N(\boldsymbol{r}'')}\right)_v$ in eq (3.2–35) and eq (3.2–36) is defined as the chemical hardness η.

doi:10.6342/NTU202403528

$$\eta = \frac{1}{2}\left(\frac{\partial^2 E}{\partial N^2}\right) \approx \frac{1}{2}\left[-\frac{2E_{EA}}{2(\int \Delta\rho_+(\boldsymbol{r})d\mathbf{s}d\mathbf{r})^2} + \frac{2E_{IP}}{2(\int \Delta\rho_-(\boldsymbol{r})d\mathbf{s}d\mathbf{r})^2}\right] = \frac{E_{IP} - E_{EA}}{2} \quad (3.2\text{–}38)$$

Now use the number of electrons in system, i.e. N, in place of density functional $\rho_N(\boldsymbol{r})$, we have:

$$\begin{aligned} E(N) &= E(N_0) + \left(\frac{\partial E}{\partial N}\right)_v (N - N_0) + \frac{1}{2}\left(\frac{\partial^2 E}{\partial N^2}\right)_v (N - N_0)^2 + \cdots \\ &= E(N_0) - \chi_m(N - N_0) + \eta(N - N_0)^2 + \cdots \end{aligned} \quad (3.2\text{–}39)$$

Here, N_0 represented the number of electrons in the unperturbed system. Systems governed by eq (3.2–39) has a saturation threshold for gaining bound electrons. This is because after saturation the addition of elections no longer influences energy $E(N)$, which implies that these excessive ones are free electrons.[97] To determine this threshold N_{me}, find the extremum of eq (3.2–39) with respect to N.

$$\left.\frac{dE(N)}{dN}\right|_{N_{me}} = -\chi_M + 2\eta(N_{me} - N_0) = 0 \quad (3.2\text{–}40)$$

$$N_{me} - N_0 = \frac{\chi_m}{2\eta} \quad (3.2\text{–}41)$$

The energy to corresponding to the maximum number of elections (N_{me}) is

$$E(N_{me}) = E(N_0) - \frac{\chi_m^2}{2\eta} + \frac{\chi_m^2}{4\eta^2} \quad (3.2\text{–}42)$$

doi:10.6342/NTU202403528

In particular, $\omega = \frac{\chi_m^2}{2\eta} = \frac{(E_{IP}+E_{EA})^2}{4(E_{IP}-E_{EA})}$ is defined as the electrophilicity index, representing an approximate energy decrease with respect to the state of bound electron saturation. The electronegativity χ_m, hardness η, and electrophilicity index ω are employed in studies on chemical reactivity, stability, reaction selectivity, reaction mechanisms, and kinetic modeling. For example, electronegativity χ_m is utilized in constructing kinetic models for radical polymerization.[186-188] Hardness η itself serves as a measure of chemical stability, characterizing the fundamental energy gap[2, 165]. Additionally, it forms the basis of HSAB theory[189] (Hard and Soft Lewis Acid and Base), which states "hard likes hard and soft likes soft" in acid-base reaction. The electrophilicity index ω measures chemical reactivity in electrophilic (i.e. electron-accepting) reactions and is valuable in understanding aromaticity, superacidity, and spectral shifts in molecular systems[167].

3.2.3. Synthetic Accessibility Score (SAscore)

The synthetic accessibility score (SAscore)[64, 168] is a metric used to evaluate the molecular structural complexity and non-usuality. From its definition, the synthetic accessibility goes from high to low as SAscore goes from 1.0 to 10.0. It is established by analyzing the molecular structural complexity and the occurrence of molecular fragments in a subset (934,064 species) of PubChem database. To calculate $SAscore(\boldsymbol{u_i})$ for molecular species $\boldsymbol{u_i}$, the raw fragment score $Score_F(\boldsymbol{u_i})$ and raw complexity score $Score_C(\boldsymbol{u_i})$ need to be calculated in advance.

Particularly, the evaluation of raw fragment score $Score_F(\boldsymbol{u_i})$ relies on a fragment scoring dictionary $Fragdict = \{(Frag_j, FragScore_j), j = 1,2, \ldots\}$, which is given in RDKit package.[64] In this dictionary, high scores are assigned to the most common

doi:10.6342/NTU202403528

fragments encountered within the PubChem subset. Conversely, fragments not found in this subset are assigned a default raw score of -4.0. The raw fragment score for molecular species $\boldsymbol{u_i}$, i.e. $Score_F(\boldsymbol{u_i})$, is determined as the averaged raw score per fragment in $\boldsymbol{u_i}$.

$$Score_F(\boldsymbol{u_i}) = \frac{\sum_k n_k(\boldsymbol{u_i})\, FragScore_k}{\sum_k n_k(\boldsymbol{u_i})} \tag{3.2–43}$$

Here, $n_k(\boldsymbol{u_i})$ is the number of $Frag_k$ occurrences in $\boldsymbol{u_i}$, which can be determined from the Morgan fingerprint calculated using RDKit. $FragScore_k = -4.0$ if $Fragdict[Frag_k] = \emptyset$. On the other hand, complexity score $Score_C(\boldsymbol{u_i})$ is determined from the number of stereo-genic centers, cyclic substructures, constituent atoms, and the number of feature types possessed by the molecular species. Here, the treatment of $Score_C(\boldsymbol{u_i})$ in RDKit is presented. This is slightly different from the original implementation.[168]

$$SizePenalty(\boldsymbol{u_i}) = nAtom(\boldsymbol{u_i})^{1.005} - nAtom(\boldsymbol{u_i}) \tag{3.2–44}$$

$$StereoPenalty(\boldsymbol{u_i}) = \log(nStereoCenters(\boldsymbol{u_i}) + 1) \tag{3.2–45}$$

$$SpiroPenalty(\boldsymbol{u_i}) = \log(nSpiroAtoms(\boldsymbol{u_i}) + 1) \tag{3.2–46}$$

$$MacroCyclePenalty(\boldsymbol{u_i}) = log[\min\big(1, nMacroCycles(\boldsymbol{u_i})\big) + 1] \tag{3.2–47}$$

$$CorrectFgpDen(\boldsymbol{u_i}) = 0.5 \ln\left[\max\left(1, \frac{nAtoms(\boldsymbol{u_i})}{nFgpFeatureTypes(\boldsymbol{u_i})}\right)\right] \tag{3.2–48}$$

$$\begin{aligned} Score_C(\boldsymbol{u_i}) = &-SizePenalty(\boldsymbol{u_i}) - StereoPenalty(\boldsymbol{u_i}) \\ &- SpiroPenalty(\boldsymbol{u_i}) - MacroCyclePenalty(\boldsymbol{u_i}) \\ &- CorrectFgpDen(\boldsymbol{u_i}) \end{aligned} \tag{3.2–49}$$

doi:10.6342/NTU202403528

Raw SAscore is defined as eq (3.2–50). If $rawSAscore(\boldsymbol{u_i}) > 8$, smooth the 10-end using natural log function, as (3.2–51) shows.

$$rawSAscore(\boldsymbol{u_i}) = 11.0 - \frac{9.0}{6.5}[Score_F(\boldsymbol{u_i}) + Score_C(\boldsymbol{u_i}) + 5.0] \quad (3.2–50)$$

$$rawSAscore(\boldsymbol{u_i}) = 8.0 + \ln[rawSAscore(\boldsymbol{u_i}) - 8.0] \quad (3.2–51)$$

Finally, scaling the $rawSAscore(\boldsymbol{u_i})$ to closed interval $[1.0, 10.0]$, obtain $SAscore(\boldsymbol{u_i})$.

$$SAscore(\boldsymbol{u_i}) = \max(\min[10.0, rawSAscore(\boldsymbol{u_i})], 1.0) \quad (3.2–52)$$

It should be noted that SAscore does not explicitly evaluate the level of difficulty in synthesizing a chemical through reactions. It is more like a metric to assess molecular structural complexity and similarity with common chemicals. Also note that raw fragment score $Score_F(\boldsymbol{u_i})$ will be -4.0 for a chemical not possessing any fragment in the fragment scoring dictionary $Fragdict$. In this scenario. $rawSAscore(\boldsymbol{u_i})$ will be greater than $\left[11.0 - \frac{9.0}{6.5}(-4.0 + 5.0)\right] = 4.08$ according to (3.2–50). It is therefore reasonable to consider $SAscore(\boldsymbol{u_i}) = 4.0$ to be a rough dividing line of synthetic accessibility.

3.2.4. Synthetic Complexity Score (SCscore)

The synthetic complexity score (SCscore)[169] is a metric used to evaluate the level of difficulty in synthesizing a particular chemical. Specifically, the level of difficult is quantified by the required number of reaction steps. From its definition, the synthetic

doi:10.6342/NTU202403528

complexity goes from low to high as SCscore goes from 1.0 to 5.0. It is established by employing a 6-layer neural network model to learn 12 million reactions from Reaxy database. Notably, during training, it is required that the product complexity should always be greater than the complexity of any of the reactant. To calculate $SCscore(\boldsymbol{u_i})$ for molecular species $\boldsymbol{u_i}$, it is necessary to prepare Morgan fingerprint $MGFgp(\boldsymbol{u_i})$ in advance. The model details are presented in **Table 3.2-3**.

Table 3.2-3. The dimensions of every layer in SCscore model.

Layer j	Weight $\boldsymbol{W_j}$	Bias $\boldsymbol{b_j}$	Normalized layer output $\boldsymbol{s_j}$ $\boldsymbol{s_j} = \text{Normalize}(\boldsymbol{s_{j-1}W_j + b_j})$
j = 1	1024×300	1×300	ReLU: let every negative entry in $(\boldsymbol{s_{j-1}W_j + b_j})$ be 0
j = 2	300×300	1×300	
j = 3	300×300	1×300	
j = 4	300×300	1×300	
j = 5	300×300	1×300	
j = 6	300×300	1×1	Softmax: $\boldsymbol{s_j} = 1.0 + \frac{4.0}{1.0 + \exp[-(\boldsymbol{s_{j-1}W_j + b_j})]}$

†$\boldsymbol{s_0} = MGFgp(\boldsymbol{u_i})$ is the fingerprint for species $\boldsymbol{u_i}$ in 1×1024 dimensions.

††The normalized output $\boldsymbol{s_6}$ is the $SCscore(\boldsymbol{u_i})$ for species $\boldsymbol{u_i}$.

Since the SAscore and SCscore evaluate different aspects of synthetic feasibility, combining them can provide more comprehensive insights. Consider a sequential reaction in which the SAscore and SCscore are evaluated for every (intermediate) product. From the perspective of SCscore, a practical sequential reaction should exhibit a monotonically increasing SCscore curve with respect to reaction steps. Therefore, if a decline in the

doi:10.6342/NTU202403528

SCscore is observed, it suggests that the reverse reaction may be more practical. Conversely, a surge in the SAscore curve indicates that rare and complex substructures, such as rings, have been formed through the reaction steps. If the SAscore significantly decreases while the SCscore significantly increases throughout the reaction steps, it is advisable to directly purchase the end product (or its precursor) from chemical suppliers.[169]

It should be emphasized that reaction steps associated with reasonable variations in SAscore and SCscore curves are not necessarily practical. To identify the most realistic reaction pathway, one may resort to computer-assisted synthesis planning (CASP)[126] software, such as AiZynthFinder[125] and ASKCOS[126]. When provided with a chemical, CASP software plans practical reaction steps for synthesizing the chemical from common precursors.

doi:10.6342/NTU202403528

3.3. Reverse Algorithm (I): MARS+ Package

3.3.1. Structure Manipulations – uni-molecular operations

The mutation operation modifies the based elements within a molecule within the context of Myelodysplastic Syndromes (MDS) research. MARS+ offers 9 distinct mutation methods: *addition*, *subtraction*, *insertion*, *element change*, *bond change*, *cyclization*, *decyclization*, *cis-trans inversion*, and *chirality inversion*. **Figure 3.3-1** illustrates each of these 9 operations. Notably, the *addition*, *subtraction*, *bond change*, and *cyclization* operations have been refined from previous versions of MARS to enhance capability, stability, and reliability. These operations are detailed below:

- **Addition**: This operation introduces a new element with a specified bond order to a molecule. The introduced element must possess a free valence compatible with the specified bond order. If either the existing molecule or the introduced element lacks the necessary free valence, the addition is cancelled. *Addition* leads to a new branch substructure, as the introduced element becomes an endpoint in the molecular graph.

- **Subtraction**: This operation removes a designated element from the molecule. The removed element's parent and descendant elements are then connected with a user-specified bond order, while preserving the remaining molecular connectivity. However, the operation is cancelled if the resulting valences of the parent or descendant elements are incompatible.

- **Insertion**: This operation introduces a new element between two connected elements in a molecule. The insertion involves replacing the existing bond between them with

doi:10.6342/NTU202403528

a specified substructure containing a central element and its connecting bond orders, i.e. “[bond_I][element][bond_II]”. Both originally connected elements must possess free valences compatible with the specified substructure for the *insertion* to proceed. Otherwise, the operation is cancelled.

- **Element Change**: This operation modifies an element and its associated bond orders with its parent and descendant elements. Essentially, it replaces a substructure of the form "[bond_I][element][bond_II]" with a new element and its compatible bond orders. Compatibility checks are performed to ensure the new element can connect appropriately with the parent and descendant elements.

- **Bond Change**: This operation modifies a bond order and the elements at its two ends, effectively replacing a substructure of the form "[element_I][bond][element_II]" with a different substructure of the same form. Similar to element change, compatibility checks are performed to ensure the new substructure can connect seamlessly with the surrounding elements.

- **Cyclization**: This operation generates a cyclic substructure with at least five member elements in the ring. This lower limit is set to avoid torsional hindrance issues commonly encountered in smaller cyclic substructures, but it can be adjusted if required. Cyclization involves labeling two designated elements with the same cyclic flag (a unique identifier) and then connecting them with a specified bond order. If either element lacks the necessary free valence for the specified bond order, the cyclization is cancelled.

doi:10.6342/NTU202403528

- **Decyclization**: This operation breaks open a cyclic substructure identified by a specific cyclic flag number. The cyclic bond order is reverted to regular bonds on the two relevant atoms. Subsequently, the remaining cyclic flags are renumbered to maintain consecutive numbering.

- **Cis-Trans Inversion**: This operation flips the cis-trans isomerism of a double bond. It essentially changes the notation in the array of cis-trans front/end flags from "\" to "/" (or vice versa).

- **Chirality Inversion**: This operation modifies the chirality of a chiral center. It flips the chirality flag from 1 (representing anti-clockwise winding) to 2 (representing clockwise winding) or vice versa. Notably, MARS+ assigns default isomerisms (trans and clockwise winding) when a potentially isomeric substructure is formed during other operations.

It is important to note that some operations may generate double-bond or triple-bond free valences. For instance, the addition operation can introduce a C(=)(-)(-) element (ID = 2) that forms a single bond with the molecule. In this scenario, the remaining (=)(-) bonds become free valences, implicitly representing attachment points for three hydrogen atoms. These free valences can be utilized in subsequent operations.

doi:10.6342/NTU202403528

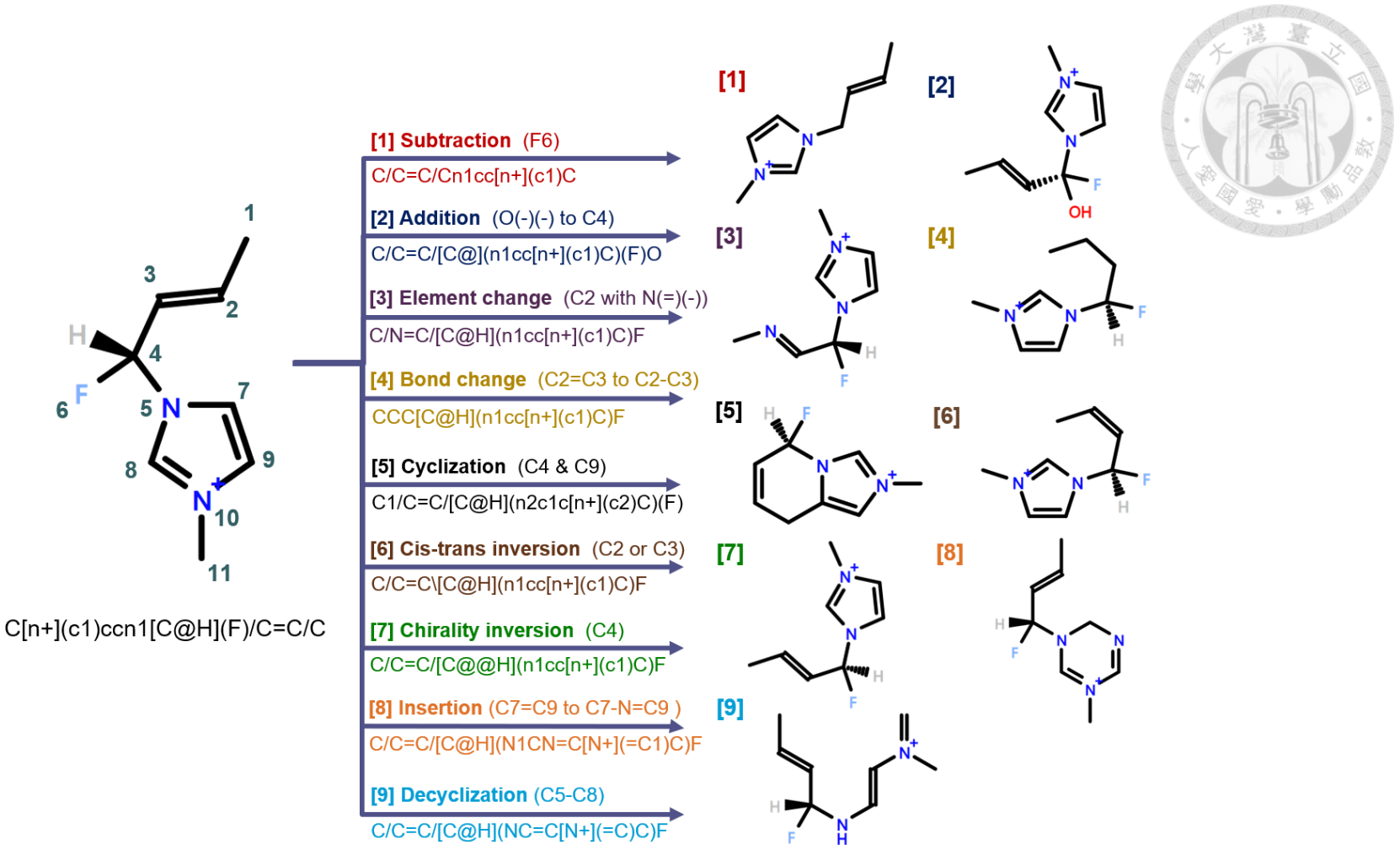


Figure 3.3-1. Illustration of the nine uni-molecular operations. Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.

doi:10.6342/NTU202403528

3.3.2. Structure Manipulations – bi-molecular operations

In addition to mutation operations, MARS+ offers functionalities for generating new molecules by combining fragments from two existing molecules or exchanging substructures between them. These bi-molecular operations provide a powerful tool for exploring chemical space. Here, we describe the two bi-molecular operations: crossover and combination.

- **Crossover**: The crossover operation mimics the biological process of chromosome crossover during meiosis. It generates new molecules by exchanging fragments between two input molecules. This operation requires specifying a bond from each of input molecule as the crossover point. If the bond orders of the designated crossover points are identical, the operation exchanges the molecular fragments based on those points. **Figure 3.3-2** exemplifies the crossover operation between [C4mim] and [P4,4,4,4], resulting in the generation of [C4C3im] and [P2,4,4,4]. If crossover leads to unpaired cyclic flags in fragments, the ring will be destructed.

- **Combination**: The combination operation merges two input molecules into a single new molecule by forming a bond between designated points on each molecule. This operation involves selecting one element from each input molecule and a free valence from each chosen element. If the bond orders of the selected free valences are the same, the operation connects the two molecules through these free valences. **Figure 3.3-2** illustrates one possible combination product of [C4mim] and [P4,4,4,4]. Notably, MARS+ allows for flexibility in choosing any single element from each molecule for the combination operation.

doi:10.6342/NTU202403528

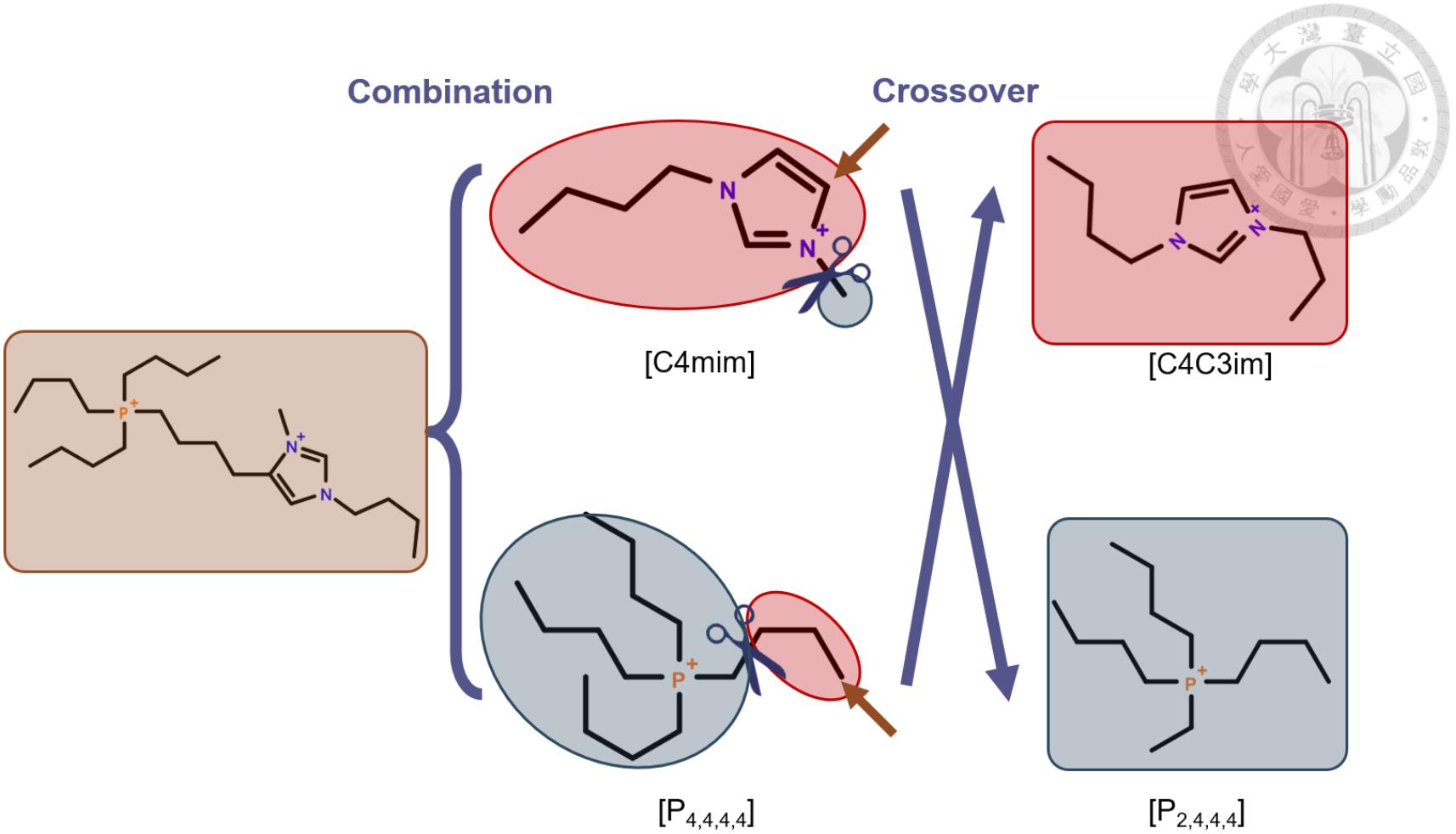


Figure 3.3-2. Illustration of the two bi-molecular operations. The crossover point is represented by the scissor symbol, while the combination point is denoted by the brown arrow. Adapted with permission from the reference[163]. Copyright 2023 American Chemical Society.

doi:10.6342/NTU202403528

3.3.3. Structure Manipulations – bi-supermolecular operations

MARS+ introduces a distinctive operation known as component swap, which differs from the uni-molecular and bi-molecular operations discussed earlier. Unlike these operations, which are aimed at generating new molecular structures, component swap delves into the combinatorial space spanned by existing chemical components, without creating novel molecular entities. This operation is pivotal in exploring diverse chemical formulations.

Essentially, component swap exchanges two specified components between two separate supermolecules. For instance, in the context of Molecular Data Structure (MDS) representation for ionic liquids, where a cation and an anion are distinct components, component swap allows for the interchange of either the cation or the anion components. **Figure 3.3-3** provides an example demonstrating an anion swap operation applied to two distinct ionic liquids.

doi:10.6342/NTU202403528

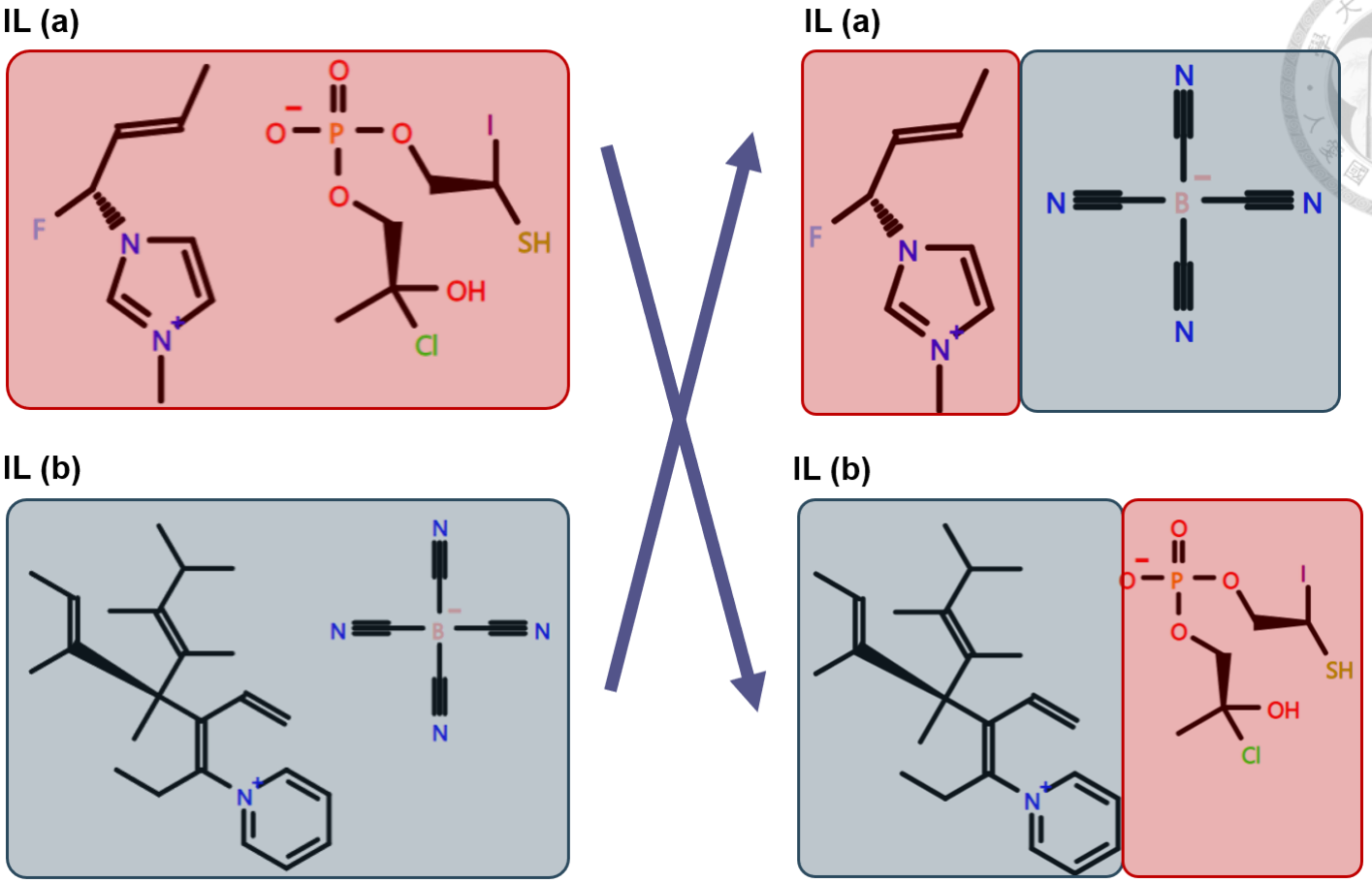


Figure 3.3-3. Illustration of the component swap operation. Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.

doi:10.6342/NTU202403528

3.3.4. Transformation of SMILES into MDS (smi2mds)

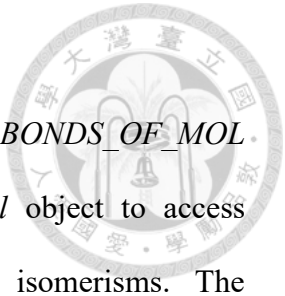
In a CAMD task, the starting chemical species can be either generated randomly from pre-defined libraries or user-specified. In particular, the latter scenario necessitates robust descriptors for transforming other molecular input formats into the MDS representation. MARS+ utilizes SMILES (Simplified Molecular Input Line Entry System) strings as the standard input and output format in its applications in CAMD tasks. To realize input transformation, MARS+ integrates functions from the OpenBabel C++ API. Here's a breakdown of the process:

- **SMILES Input and Conversion**: The input SMILES string is stored in a variable named *SMILES_stringstream*. The *smi2mds_OBabel()* function utilizes the OpenBabel conversion object (*OBConversion*) to transform the SMILES string into a 3D Open Babel Molecular object (*OBMol*).

- **Enriching the OBMol Object**: Initially, the OBMol object lacks hydrogen atoms and atomic coordinates. The *AddHydrogens()* function adds implicit hydrogen atoms to the molecule.The *Build()* function calculates 3D coordinates for each atom. These steps are crucial for accurate isomerism perception.

- **Isomerism Detection**: The *HasCisTransStereo()* function checks for the presence of cis/trans isomers in the molecule. The *HasTetrahedralStereo()* function checks for the presence of tetrahedral stereocenters (potential chiral centers). If isomers are detected, detailed information can be retrieved using *GetCisTransStereo()* and *GetTetrahedralStereo()* functions.

doi:10.6342/NTU202403528



- **Data Extraction and MDS Conversion**: Loops like *FOR_BONDS_OF_MOL* and *FOR_ATOMS_OF_MOL* iterate through the *OBMol* object to access information about bonds, atom types, and detected isomerisms. The *smi2mds_OBabel()* function, defined in the *src/MOLECULE.cpp* file, utilizes this extracted data to construct the corresponding MDS for the molecule.

For a more in-depth explanation of the technical details, refer to **Algorithm 1** in Appendix A.

3.3.5. Transformation of MDS into SMILES (mds2smi())

MARS+ offers the *mds2smi()* function (defined in *src/MOLECULE.cpp*) to convert a MDS to a SMILES string. This conversion process relies on two helper matrices: *Bindex* and *atomsmi*. *Bindex* is a two-dimensional matrix where each row *Bindex[i]* corresponds to the *(i+1)-th* element in the MDS. Initially, *Bindex[i]* is a copy of the bond orders defined for that element in the MDS (section 3.1.1). As the conversion progresses, for each *(j+1)-th* valence of element *i* used to form a bond with another element within the molecule, the corresponding entry in *Bindex[i]* (i.e., *Bindex[i][j]*) is set to 0. Consequently, the remaining non-zero values in *Bindex[i]* represent the element's free valences, which are assumed to be connected to implicit hydrogen atoms in the SMILES output.

Similarly, *atomsmi* is a two-dimensional matrix where each row *atomsmi[i]* maps the SMILES of the *(i+1)-th* element to proper output positions in the overall molecular SMILES string, so as to form a valid SMILES. This mapping is determined by the molecular connectivity specified by *Bindex*. For instance, in the case of ethane (canonical

doi:10.6342/NTU202403528

SMILES: CC), the positions of C characters in the output SMILES string depend on how the carbons are connected, a process detailed in **Algorithm 2** in Appendix A.

Once the raw SMILES representation is generated with the aid of *atomsmi*, it can be canonicalized using the Open Babel API. Moreover, Open Babel supports the conversion of SMILES into various other chemoinformatic formats such as MOLfile and GJFfile, broadening the utility and compatibility of the output data.

3.4. Reverse Algorithm (II): Selection Algorithms

3.4.1. Fitness Function

As mentioned in section 2.1, the optimality of a designed species $\boldsymbol{u_i}$ (at thermodynamic state $\boldsymbol{w_i}$) is characterized from fitness function $Fitfcn(\boldsymbol{u_i}, \boldsymbol{w_i}; \boldsymbol{t})$ or objective function $Objfcn(\boldsymbol{u_i}, \boldsymbol{w_i}; \boldsymbol{t})$ in view of the mathematical framework. In this work, we choose to use fitness function $Fitfcn(\boldsymbol{u_i}, \boldsymbol{w_i}; \boldsymbol{t})$, which by its significance should give higher score as the molecular properties are closer to the target values. The most straightforward way to formulate a fitness functions is to utilize the discrepancy between predicted molecular properties $\boldsymbol{f}(\boldsymbol{z}(\boldsymbol{u_i}), \boldsymbol{w_i}) = [f_1(\boldsymbol{z}(\boldsymbol{u_i}), \boldsymbol{w_i}); \dots; f_{p+q}(\boldsymbol{z}(\boldsymbol{u_i}), \boldsymbol{w_i})]$ and the properties specifications $\boldsymbol{t} = [t_1, \dots, t_n]$ (see section 2.1 for notations). For convenience, the absolute deviation of molecular property j from the corresponding target value is denoted as $\Delta_j(\boldsymbol{u_i}, \boldsymbol{w_i}; \boldsymbol{t}) = \left| f_j(\boldsymbol{z}(\boldsymbol{u_i}), \boldsymbol{w_i}) - t_j \right|$. The averaged absolute deviation over the n properties is denoted as $\langle \Delta_j(\boldsymbol{u_i}, \boldsymbol{w_i}; \boldsymbol{t}) \rangle = \sum_j^n \Delta_j(\boldsymbol{u_i}, \boldsymbol{w_i}; \boldsymbol{t}) \,/n$.

Note that it is often difficult to predict whether there will be at least a designed chemical species $\boldsymbol{u_i}$ that makes $\Delta_j(\boldsymbol{u_i}, \boldsymbol{w_i}; \boldsymbol{t})$ go infinity during the optimization process, as it depends on the nature of the chemical species $\mathbf{u_i}$ and the property

doi:10.6342/NTU202403528

estimation method f_j for each of the property j. For this robustness issue, it is the best practice to devise a well-behaved and finite-bounded fitness function that can map $\Delta_j(\boldsymbol{u_i}, \boldsymbol{w_i}; \boldsymbol{t})$ to a finite value even when $\Delta_j(\boldsymbol{u_i}, \boldsymbol{w_i}; \boldsymbol{t})$ is infinite. For this purpose, the mathematical form we use is:

$$Fitfcn(\boldsymbol{u_i}, \boldsymbol{w_i}; \boldsymbol{t}) = A - \frac{B}{1.0 + C \exp\left(\frac{-\langle \Delta_j(\boldsymbol{u_i}, \boldsymbol{w_i}; \boldsymbol{t}) \rangle}{D}\right)} \tag{3.4–1}$$

Here, A, B, and C are positive coefficients that determine the higher bound (eq. (3.4–2)) and lower bound (eq. (3.4–3)) of fitness, and D is a positive coefficient that affects the decaying rate of fitness with respect to $\langle \Delta_j(\boldsymbol{u_i}, \boldsymbol{w_i}; \boldsymbol{t}) \rangle$.

$$Fitfcn_{hbnd} = A - \frac{B}{1.0 + C} \tag{3.4–2}$$

$$Fitfcn_{lbnd} = A - B \tag{3.4–3}$$

These parameters are empirically set as $A = 6.0$, $B = 5.0$, $C = 4.0$, and $D = 3.0$, with $Fitfcn_{hbnd} = 5.0$ and $Fitfcn_{lbnd} = 1.0$. The decay of fitness with respect to $\langle \Delta_j(\boldsymbol{u_i}, \boldsymbol{w_i}; \boldsymbol{t}) \rangle$ is shown in **Figure 3.4-1**.

doi:10.6342/NTU202403528

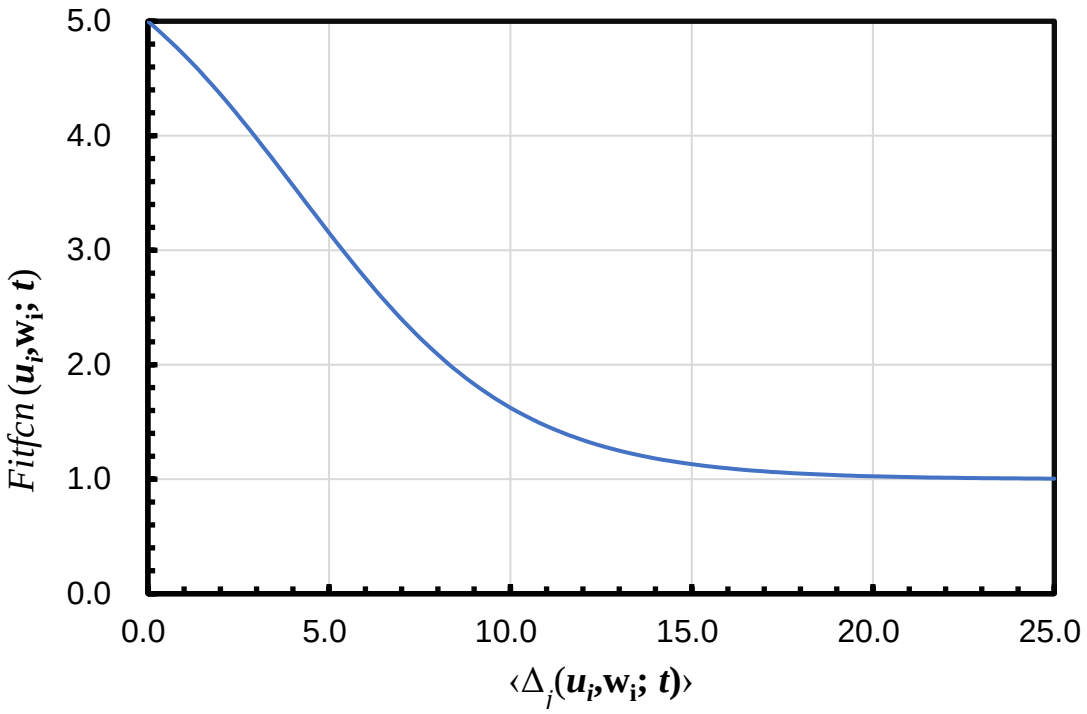


Figure 3.4-1. The relationship between fitness (eq. (3.4–1)) and the mean deviation of properties from targets $\langle \Delta_j(\mathrm{m_i}, \mathrm{s_i}; t) \rangle$ with parameter A=6, B=5, C=4, and D=3.

3.4.2. Selection Algorithms

Let N_{popu} be the number of chemical entities at the current iteration n, denoted by the set $Popu_n = \left\{(\boldsymbol{u_1^n}, \boldsymbol{w_1^n}), \ldots, \left(\boldsymbol{u_{N_{popu}}^n}, \boldsymbol{w_{N_{popu}}^n}\right)\right\}$, where $\boldsymbol{u}$ represents chemical structure and $\boldsymbol{w}$ represents thermodynamics state. A selection algorithm $\boldsymbol{P}$ is employed to sample a specific number of entities from $Popu_n$. This sample, denoted by the set Sel_n, may contain duplicates due to the possibility of repeated selections. Subsequently, genetic operators are applied to the chemicals within Sel_n to generate new chemical entities. The collection of these newly generated chemicals is denoted by Gen_n. We have implemented several selection algorithms $\boldsymbol{P}$ to select chemicals from $(Popu_n \cup Gen_n)$ as the subject chemicals in the next iteration. In other words, $\boldsymbol{P}(Popu_n \cup Gen_n) = \{(\boldsymbol{u_a^n}, \boldsymbol{w_a^n}), (\boldsymbol{u_b^n}, \boldsymbol{w_b^n}), (\boldsymbol{u_c^n}, \boldsymbol{w_c^n}), \ldots\} = Popu_{n+1}$.

doi:10.6342/NTU202403528

- **New-species Roulette Wheel (RW)**

This scheme only selects from the newly generated chemicals, i.e. $\boldsymbol{P}(Gen_n) = \{(\boldsymbol{u_a^n}, \boldsymbol{w_a^n}), (\boldsymbol{u_b^n}, \boldsymbol{w_b^n}), (\boldsymbol{u_c^n}, \boldsymbol{w_c^n}), \dots\} = Popu_{n+1}$. The probability for a chemical to be selected to the next iteration is determined by the fraction of its fitness in Gen_n.

$$P^{RW}(\boldsymbol{u_i^n}, \boldsymbol{w_i^n}; \boldsymbol{t}) = \frac{Fitfcn(\boldsymbol{u_i^n}, \boldsymbol{w_i^n}; \boldsymbol{t})}{\sum_j^{Gen_n} Fitfcn(\boldsymbol{u_j^n}, \boldsymbol{w_j^n}; \boldsymbol{t})} \tag{3.4–4}$$

- **Linearly-scaled Individual Fitness (LSIF)**

Similar to RW, this scheme only selects from the newly generated chemicals, i.e. $\boldsymbol{P}(Gen_n) = \{(\boldsymbol{u_a^n}, \boldsymbol{w_a^n}), (\boldsymbol{u_b^n}, \boldsymbol{w_b^n}), (\boldsymbol{u_c^n}, \boldsymbol{w_c^n}), \dots\} = Popu_{n+1}$. The probability for a chemical to be selected to the next iteration is determined solely by its normalized fitness. The normalization constant is the higher bound of the fitness function, i.e. eq (3.4–2).

$$P^{LSIF}(\boldsymbol{u_i^n}, \boldsymbol{w_i^n}; \boldsymbol{t}) = \frac{Fitfcn(\boldsymbol{u_i^n}, \boldsymbol{w_i^n}; \boldsymbol{t})}{A - \frac{B}{1.0 + C}} \tag{3.4–5}$$

- **Simulated Annealing (SA)**

We called it a child chemical when the chemical is generated from applying genetic operators to its parent chemicals. This scheme compares a child chemical $(\boldsymbol{x_i^n}, \boldsymbol{y_i^n}) \in Gen_n$ with its parent chemicals $(\boldsymbol{u_i^n}, \boldsymbol{w_i^n}) \in Popu_n$, and determines either of them to be selected to the next iteration, i.e. $\boldsymbol{P}(Popu_n \cup Gen_n) = \{(\boldsymbol{u_a^n}, \boldsymbol{w_a^n}), (\boldsymbol{x_b^n}, \boldsymbol{y_b^n}), \dots\} = Popu_{n+1}$. The probability for a child chemical $(\boldsymbol{x_i^n}, \boldsymbol{y_i^n}) \in Gen_n$ to be selected to the next iteration is determined by a temperature parameter T and the difference of fitness

doi:10.6342/NTU202403528

between its parent $(\boldsymbol{u_i^n}, \boldsymbol{w_i^n}) \in Popu_n$ and it.

$$P^{SA}(\boldsymbol{x_i^n}, \boldsymbol{y_i^n}; \boldsymbol{t}) = \exp\left(-\frac{Fitfcn(\boldsymbol{u_i^n}, \boldsymbol{w_i^n}; \boldsymbol{t}) - Fitfcn(\boldsymbol{x_i^n}, \boldsymbol{y_i^n}; \boldsymbol{t})}{T}\right) \quad (3.4\text{–}6)$$

In its execution, the temperature parameter T should be initialized with a sufficiently high positive value. This makes $\lim_{T\to\infty} P^{SA}(\boldsymbol{x_i^n}, \boldsymbol{y_i^n}; \boldsymbol{t}) = 1$ even child chemical $(\boldsymbol{x_i^n}, \boldsymbol{y_i^n}) \in Gen_n$ is much worse than its parent chemicals $(\boldsymbol{u_i^n}, \boldsymbol{w_i^n}) \in Popu_n$, thereby encouraging the exploration of chemical space (feasible region). As iterations proceed, the temperature is annealed using a programmed strategy, say, $T_{n+1} = \alpha T_n, 0 < \alpha < 1$. As the temperature gradually becomes lower, it becomes less likely to select a worse child species over the its (relative better) parent species. This can be seen from the fact that any nonzero positive differences $\left[Fitfcn(\boldsymbol{u_i^n}, \boldsymbol{w_i^n}; \boldsymbol{t}) - Fitfcn(\boldsymbol{x_i^n}, \boldsymbol{y_i^n}; \boldsymbol{t})\right)]$ makes $\lim_{T\to 0} P^{SA}(\boldsymbol{x_i^n}, \boldsymbol{y_i^n}; \boldsymbol{t}) = 0$.

- **Fitness Monte Carlo (FMC)**

This scheme compares a child chemical $(\boldsymbol{x_i^n}, \boldsymbol{y_i^n}) \in Gen_n$ with its parent chemicals $(\boldsymbol{u_i^n}, \boldsymbol{w_i^n}) \in Popu_n$, and determines either of them to be selected to the next iteration, i.e. $\boldsymbol{P}(Popu_n \cup Gen_n) = \{(\boldsymbol{u_a^n}, \boldsymbol{w_a^n}), (\boldsymbol{x_b^n}, \boldsymbol{y_b^n}), \ldots\} = Popu_{n+1}$. It resembles SA in mechanism, but it has no temperature parameter. After sufficient iterations using FMC, we expect the frequency distribution of all the chemicals ($TotPopu_n = \{Popu_1 \cup Popu_2 \cup \ldots \cup Popu_n\}$) against property-target discrepancy $\Delta_j(\boldsymbol{u_i}, \boldsymbol{w_i}; \boldsymbol{t})$ should be similar to $Fitfcn(\boldsymbol{u_i}, \boldsymbol{w_i}; \boldsymbol{t})$. (Recall section 3.4.1)

doi:10.6342/NTU202403528

$$P^{FMC}(\boldsymbol{x_i^n}, \boldsymbol{y_i^n}; \boldsymbol{t}) = \frac{Fitfcn(\boldsymbol{x_i^n}, \boldsymbol{y_i^n}; \boldsymbol{t})}{Fitfcn(\boldsymbol{u_i^n}, \boldsymbol{w_i^n}; \boldsymbol{t})} \quad (3.4\text{–}7)$$

- **Non-dominated Sorting (NS)**

Non-dominated sorting, based on Pareto optimality [190], is particularly useful for multi-objective optimization problem. To illustrate the concept of Pareto optimality, let us consider a population of only four chemical species $Popu = \{(\boldsymbol{u_1}, \boldsymbol{w_1}), (\boldsymbol{u_2}, \boldsymbol{w_2}), (\boldsymbol{u_3}, \boldsymbol{w_3}), (\boldsymbol{u_4}, \boldsymbol{w_4})\}$. Each $(\boldsymbol{u_i}, \boldsymbol{w_i})$ has m properties $\boldsymbol{f}(\boldsymbol{u_i}, \boldsymbol{w_i}) = [f_1(\boldsymbol{u_i}, \boldsymbol{w_i}), \dots, f_m(\boldsymbol{u_i}, \boldsymbol{w_i})]^{\mathrm{T}}$, where f_j is the model used for predicting property j. Let the fitness function evaluate every single property, i.e. $\boldsymbol{Fitfcn}(\boldsymbol{u_i}, \boldsymbol{w_i}; \boldsymbol{t}) = [Fitfcn_1(\boldsymbol{u_i}, \boldsymbol{w_i}; \boldsymbol{t}), \dots, Fitfcn_m(\boldsymbol{u_i}, \boldsymbol{w_i}; \boldsymbol{t})]^{\mathrm{T}}$, instead of lumping properties together.

For pair of chemicals $(\boldsymbol{u_i}, \boldsymbol{w_i}), (\boldsymbol{u_j}, \boldsymbol{w_j}) \in Popu$, we say $(\boldsymbol{u_i}, \boldsymbol{w_i})$ *dominates* $(\boldsymbol{u_j}, \boldsymbol{w_j})$ if the two conditions are satisfied:

(I) $\boldsymbol{Fitfcn}(\boldsymbol{u_i}, \boldsymbol{w_i}; \boldsymbol{t}) \geq \boldsymbol{Fitfcn}(\boldsymbol{u_j}, \boldsymbol{w_j}; \boldsymbol{t})$

(II) For at least a property k, $Fitfcn_k(\boldsymbol{u_i}, \boldsymbol{w_i}; \boldsymbol{t}) > Fitfcn_k(\boldsymbol{u_j}, \boldsymbol{w_j}; \boldsymbol{t})$.

In other words, $(\boldsymbol{u_i}, \boldsymbol{w_i})$ *dominates* $(\boldsymbol{u_j}, \boldsymbol{w_j})$ by improving at least one of $(\boldsymbol{u_j}, \boldsymbol{w_j})$'s property without sacrificing $(\boldsymbol{u_j}, \boldsymbol{w_j})$'s optimality in any other property. This is illustrated in **Figure 3.4-2**. When $(\boldsymbol{u_3}, \boldsymbol{w_3})$ moves horizontally to the position of $(\boldsymbol{u_1}, \boldsymbol{w_1})$, it improves $Fitfcn_2(\boldsymbol{u_i}, \boldsymbol{w_i}; \boldsymbol{t})$ without sacrificing $Fitfcn_1(\boldsymbol{u_i}, \boldsymbol{w_i}; \boldsymbol{t})$. Therefore, $(\boldsymbol{u_3}, \boldsymbol{w_3})$ is dominated by $(\boldsymbol{u_1}, \boldsymbol{w_1})$. Moving $(\boldsymbol{u_1}, \boldsymbol{w_1})$ to the position of point 2 improves property $Fitfcn_2(\boldsymbol{u_i}, \boldsymbol{w_i}; \boldsymbol{t})$ at the sacrifice of $Fitfcn_1(\boldsymbol{u_i}, \boldsymbol{w_i}; \boldsymbol{t})$. Therefore, there are not dominance relations between them.

doi:10.6342/NTU202403528

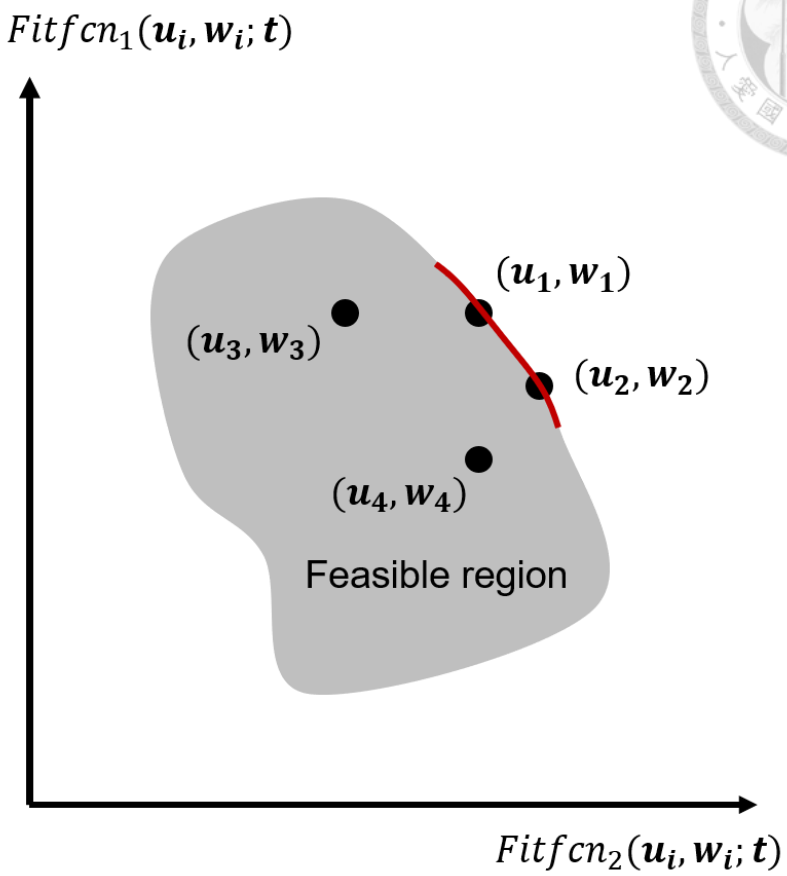


Figure 3.4-2. Schematic diagram for Pareto frontier. The number of properties is reduced to two ($m = 2$) for illustration.

After exhaustive comparisons between every pair of species in $Popu$, one may find several species not *dominated* by any others. These species are optimal, and forms a set called the first Pareto frontier, $Front_1$. Picking out these species from $Popu$, the same method can be applied to determine the second Pareto frontier, $Front_2$, and so on. The chemical species in the same $Front$ are sorted in descending order of the crowded distance.[191] The crowded distance quantifies the sparsity around a point in multi-dimensional property space. The lesser crowded solutions are preferred due to its potential for leading to wider exploration.

doi:10.6342/NTU202403528

The calculation of crowded distance is exemplified by **Table 3.4-1**. The crowded distance $\boldsymbol{u_1}$ in property *i* is determined by sorting the population in the ascending order of property *i*, followed by calculating the difference of single-property fitness between the two adjacent chemicals of $\boldsymbol{u_1}$. The crowded distance of each property sums up the overall crowded distance.

Table 3.4-1. Calculation of crowded distance for chemical $\boldsymbol{u_1}$.

Property	Sorted in ascending order of $f_j(\boldsymbol{u_i}, \boldsymbol{w_i})$				Crowded distance (CD) of $\boldsymbol{u_1}$ in property $f_j(\boldsymbol{u_i}, \boldsymbol{w_i})$
$f_1(\boldsymbol{u_i}, \boldsymbol{w_i})$	u_2	u_4	$\boldsymbol{u_1}$	u_3	$\mathrm{CD}_1(\boldsymbol{u_1}, \boldsymbol{w_1}) = Fitfcn_1(\boldsymbol{u_3}, \boldsymbol{w_3}) - Fitfcn_1(\boldsymbol{u_4}, \boldsymbol{w_4})$
$f_2(\boldsymbol{u_i}, \boldsymbol{w_i})$	u_2	$\boldsymbol{u_1}$	u_4	u_3	$\mathrm{CD}_2(\boldsymbol{u_1}, \boldsymbol{w_1}) = Fitfcn_2(\boldsymbol{u_4}, \boldsymbol{w_4}) - Fitfcn_2(\boldsymbol{u_2}, \boldsymbol{w_2})$
...	...	...	...	...	....
$f_m(\boldsymbol{u_i}, \boldsymbol{w_i})$	u_4	u_3	$\boldsymbol{u_1}$	u_2	$\mathrm{CD}_m(\boldsymbol{u_1}, \boldsymbol{w_1}) = Fitfcn_m(\boldsymbol{u_2}, \boldsymbol{w_2}) - Fitfcn_m(\boldsymbol{u_3}, \boldsymbol{w_3})$
Overall crowded distance for $\boldsymbol{u_1}$, $\mathrm{CD}(\boldsymbol{u_1}, \boldsymbol{w_1}) = \sum_{i=1}^{m} \mathrm{CD_i}(\boldsymbol{u_1}, \boldsymbol{w_1})$					

†This table illustrates the scenario of a 4-species population, $Popu = \{\boldsymbol{u_1}, \boldsymbol{u_2}, \boldsymbol{u_3}, \boldsymbol{u_4}\}$.

doi:10.6342/NTU202403528

Chapter 4. Intrinsic Performance of MARS+ based CAMD

4.1. Exhaustive Structure Operations on Every Possible Point

A key advantage of MARS+ is its transparency in comprehensively generating all possible new chemicals for each molecular operation. To illustrate this capability, we will utilize two ionic liquids depicted in **Figure 4.1-1** to exemplify the generation of all possible new molecules for each of the twelve developed operations. It is important to note that the protection mechanism is only activated for the charged atoms. For clarity in the following text, the ionic liquid in **Figure 4.1-1(a)** will be designated as *IL (a)*, with its cation and anion components referred to as *cation (a)* and *anion (a)*, respectively. Similarly, *IL (b)*, *cation (b)*, and *anion (b)* will denote the ionic liquid presented in **Figure 4.1-1(b)**.

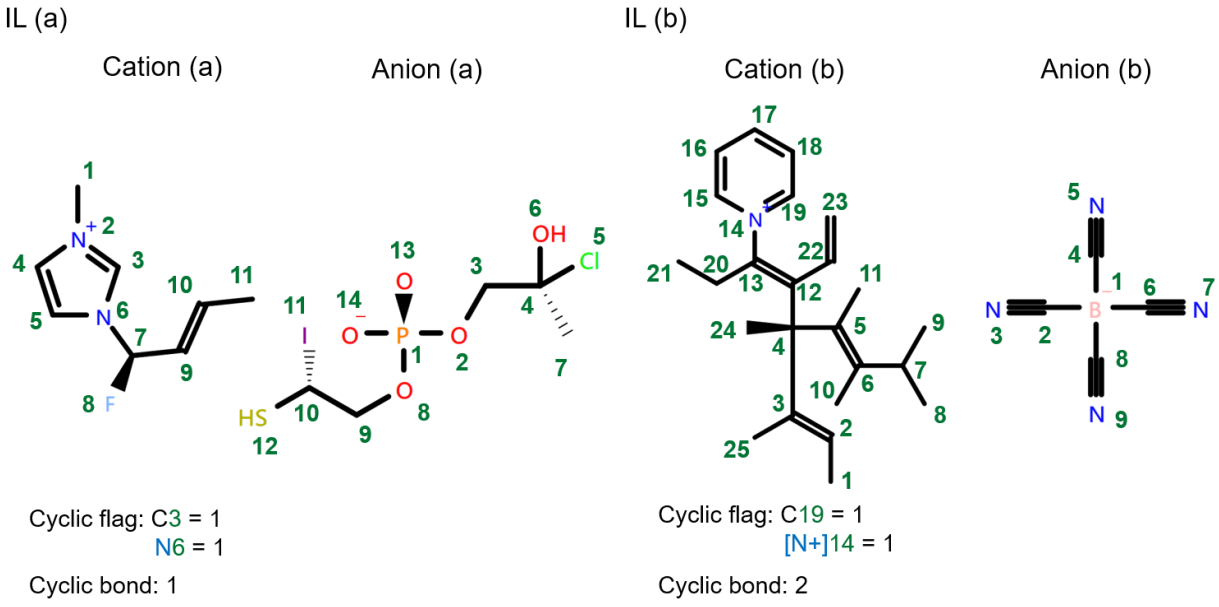


Figure 4.1-1. Exemplary ionic liquids (a) and (b) used for demonstrating the test of exhaustive single operation. Reprinted with permission from the reference[163]. Copyright

doi:10.6342/NTU202403528

4.1.1. Insertion

In the molecular data structure, any two connected elements must be linked by a bond. We denote this substructure as "[element_1][bond][element_2]". When an insertion operation occurs, a new substructure '[bond_I][element][bond_II]' replaces the existing bond. Consequently, the resulting substructure becomes "[element_1][bond_I][element][bond_II][element_2]". Notably, the newly introduced element (denoted as [element]) connects to [element_1] via [bond_I] and to [element_2] via [bond_II]. For *cation (a)*, 7 allowable bonds exist for insertion operations: C4=C5, C5-N6, N6-C7, C7-F8, C7-C9, C9=C10, and C10-C11. **Figure 4.1-2** illustrates the 25 unique cations generated out of the 31 possible combinations.

doi:10.6342/NTU202403528

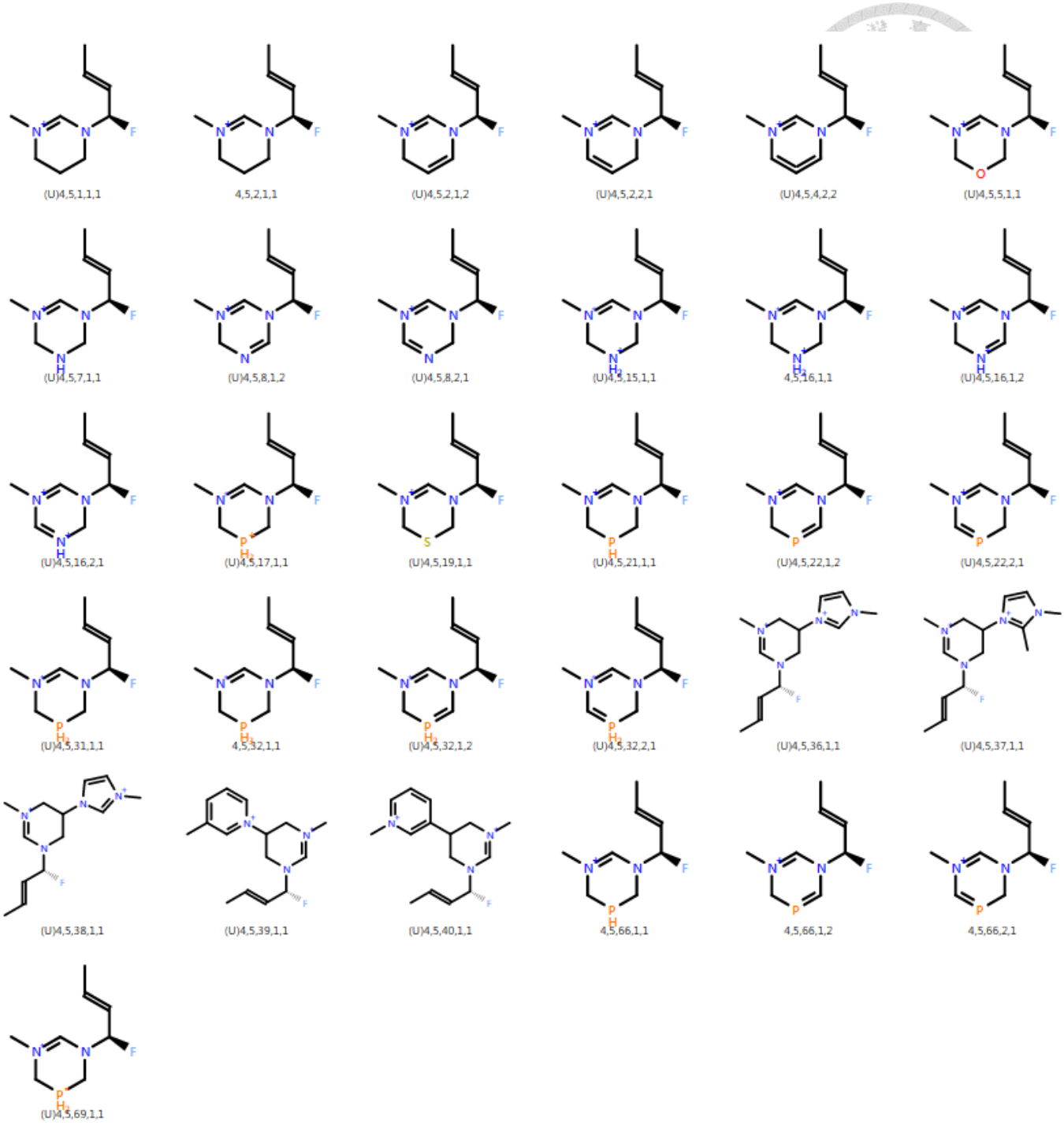


Figure 4.1-2. The 31 result cations produced from *insertion* operation on the double bonds between the 4th and the 5th element of *cation (a)*. (U) denotes a unique species among the cations shown here. (Caption: element index of element I, element index of element II, ID of the introduced element, bond order of the introduced element with the parent element, bond order of the introduced element with the descendant element).

doi:10.6342/NTU202403528

Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.

4.1.2. Cyclization

The cyclization operation in *cation (a)* involves pairing elements with single bond free valences to form rings within the molecule. Among the 8 elements eligible (*element index* = 1, 3, 4, 5, 7, 9, 10, 11), the minimum ring size allowed is set to 5 members. However, not all pairs of these elements can successfully form rings meeting this size criterion. Despite the requirement for rings to be larger than 5 members, the generated cations exhibit some smaller rings due to variations in ring perception algorithms. This includes the identification of the largest set of smallest rings (LSSR)[192], smallest set of smallest rings (SSSR)[193-195], or other ring sets. It's noted that the rings perceived by MARS's built-in algorithm often do not strictly adhere to LSSR or SSSR principles.

In future developments, enhancing algorithms for determining SSSR and implementing minimum cycle basis (MCB)[194] methodologies would be beneficial. This improvement would ensure more accurate and consistent identification of smallest ring systems within generated cations, contributing to enhanced precision in molecular structure analysis and design.

doi:10.6342/NTU202403528

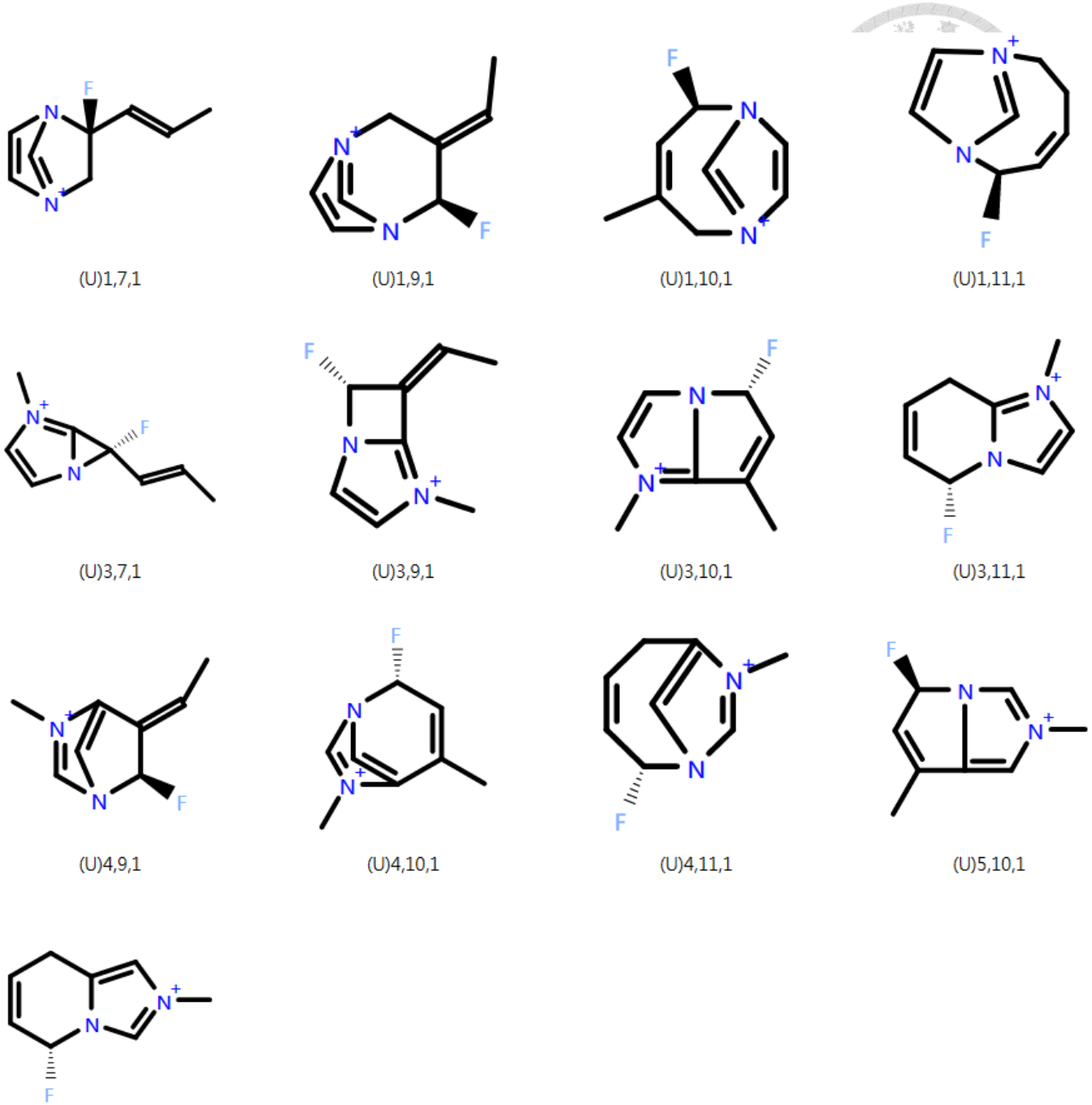


Figure 4.1-3. The 13 result cations produced from *cyclization* operation on *cation (a)*. (U) means a unique species among the cations shown here. (Caption: element index of element I, element index of element II, cyclic bond order in between). Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.

doi:10.6342/NTU202403528

4.1.3. Decyclization

The decyclization operation removes paired ring numbers and restores the cyclic bond order to the two relevant atoms. As these cyclic bond orders become free valences, they are assumed to connect with implicit hydrogen atoms. In the case of *cation (a)*, the only available point for this operation is C3-N6. The resulting cation is depicted in **Figure 4.1-4**.

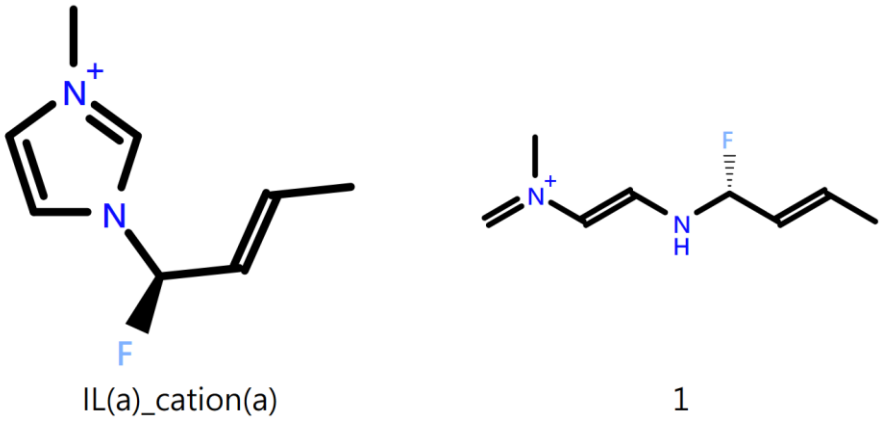


Figure 4.1-4. The structure of *cation (a)* before and after the destruction of C3-N6 ring bond (cyclic flag = 1). Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.

4.1.4. Cis-trans inversion

This operation applies to any element with either a cis-trans front flag or a cis-trans end flag. Taking *cation (a)* as an example, if we invert the cis-trans front flag of the 9th element (changing “/” to "\"), it results in a cis isomer, as depicted in Figure 4.1-5. Interestingly, altering the cis-trans end flag of the 10th element (also from “/” to "\") leads to the same structure.

doi:10.6342/NTU202403528

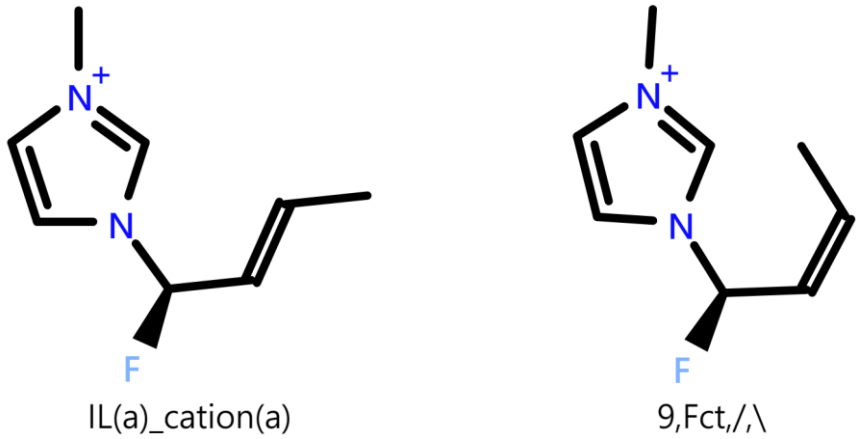
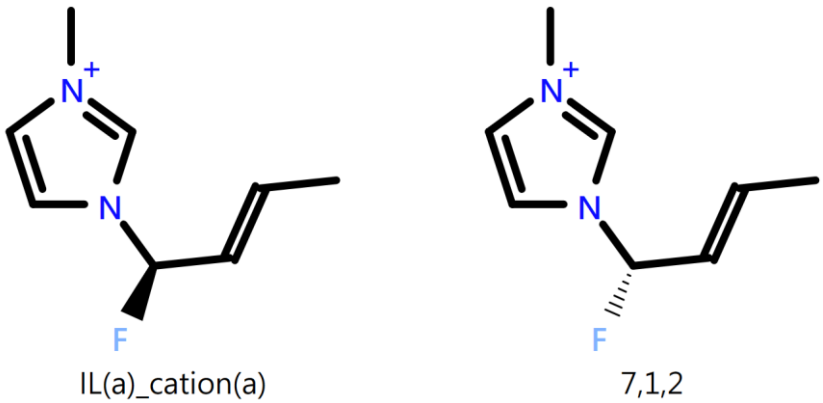


Figure 4.1-5. The structure of *cation (a)* before and after the inversion of cis-trans isomerism of the 9th element (Caption: element index of the subject element, flag type of the subject element, flag before inversion, flag after inversion) Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.

4.1.5. Chirality inversion

The *chirality inversion* operation can be applied to any chiral center within a molecule. In the case of cation (a), only the 7th element (with ID=1, represented as C(-)(-)(-)(-)) exhibits chirality. By performing the chirality inversion operation, the chirality flag of the 7th element changes from 1 (anti-clockwise winding) to 2 (clockwise winding).



doi:10.6342/NTU202403528

Figure 4.1-6. The structure of *cation (a)* before and after the inversion of cis-trans isomerism of the 7th element. (Caption: element index of chiral center, chirality flag before operation, chirality flag after operation) Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.

4.1.6. Crossover

The crossover operation facilitates the creative combination of fragments from two parent molecules to generate novel "child" molecules. This process entails the selection of a bond (crossover point) from each parent molecule. Successful crossover hinges on matching bond orders at both chosen points. When this condition is met, the molecular fragments beyond those points undergo reciprocal exchange, resulting in the formation of two structurally distinct offspring.

Since this work aims to design ionic liquids, viable crossover points are restricted to those that will lead to two positively charged child molecules. **Figure 4.1-7** demonstrates the scenario where the double bond between the 4th and 5th elements of *cation (a)* (C4=C5) serves as one of the designated crossover points. *Cation (b)* presents four potential double bonds (C5=C6, C15=C16, C17=C18, and C22=C23) that can function as the other crossover point. However, bonds such as C2=C3 and C12=C13 in cation (b) are excluded as viable options since they will result in the formation of neutral species, deviating from the desired outcome.

It is noteworthy that the ring bond, like [N+]14=C19 in *cation (b)*, cannot be the subject of crossover. This suggests limitations in their current handling by MARS+. The ring-open algorithm[110, 196] in during crossover is an ongoing development within MARS+ to address complex crossover scenarios involving cyclic structures.

doi:10.6342/NTU202403528

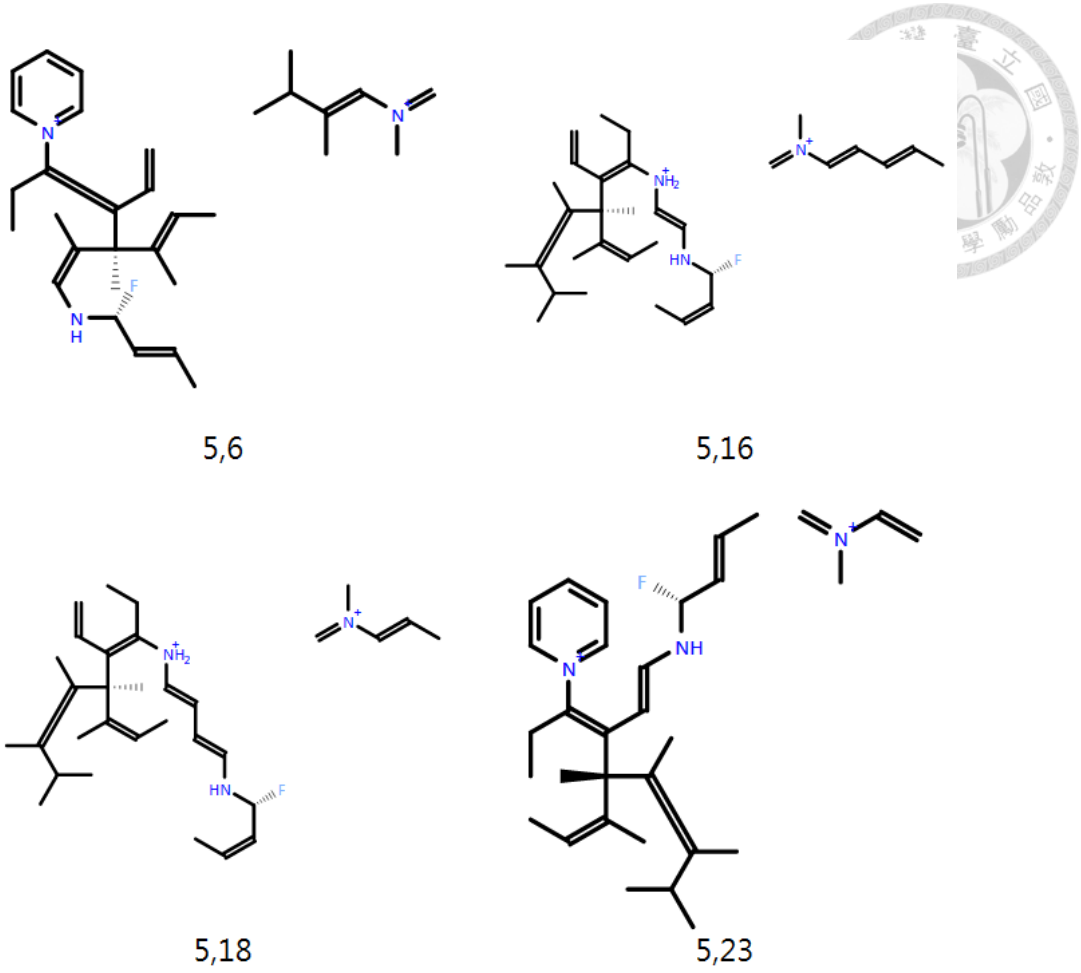


Figure 4.1-7. The 4 pairs of cations generated from applying *crossover* operation on the *cation (a)* and *cation (b)*, with crossover point of *cation (a)* fixed at the double bond between its 4th and 5th element. (Caption: crossover point for *cation (a)*, crossover point for *cation (b)*) Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.

4.1.7. Combination

The combination of two molecules is feasible when compatible free valences exist between them. In *cation (a)*, eight elements possess free single bonds: those with element index 1, 3, 4, 5, 7, 9, 10, and 11. *Cation (b)* exhibits a higher number of elements with single bond valences – eighteen in total. These elements correspond to indices 1, 2, 7, 8,

doi:10.6342/NTU202403528

9, 11, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, and 25. **Figure 4.1-8** illustrates results where the third element of cation (a) is selected as one of the potential combination points.

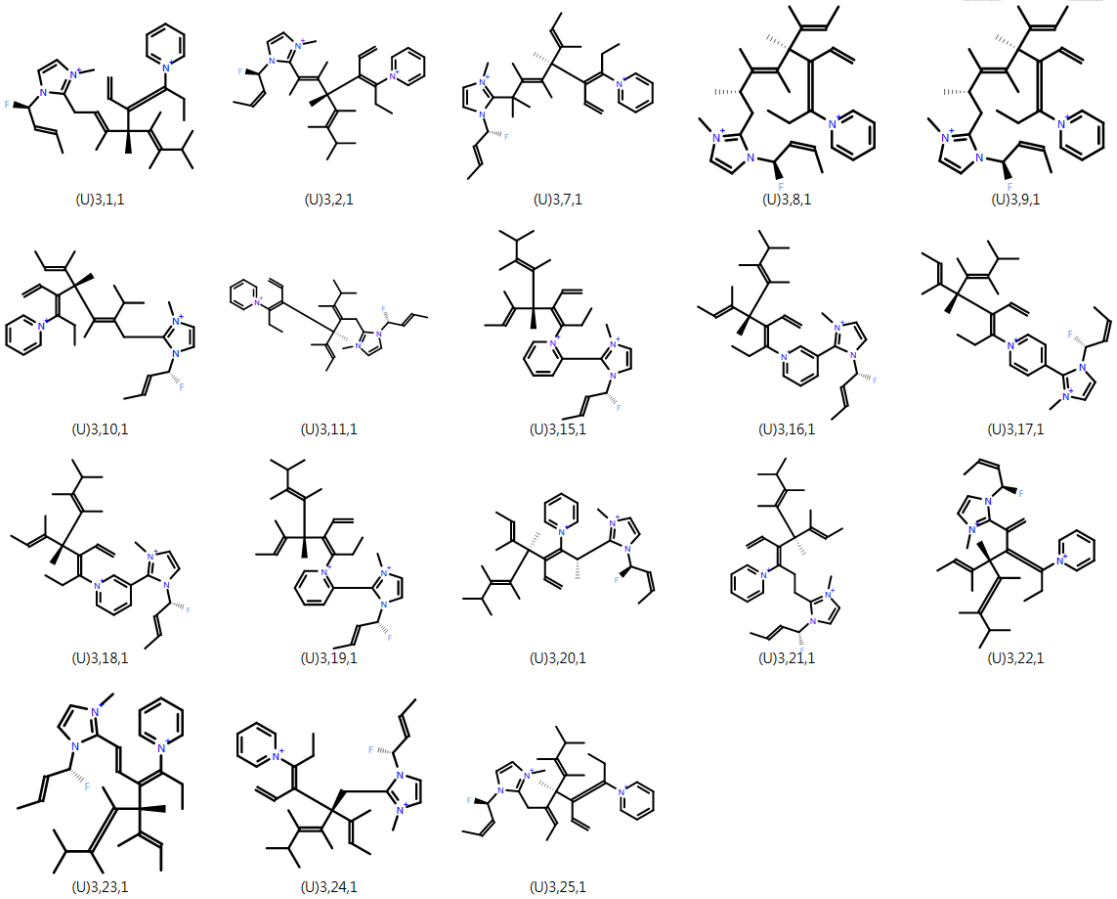


Figure 4.1-8. The 18 cations generated from applying *combination* operation on *cation (a)* and *cation (b)*, with the 3rd element of *cation (a)* picked as the combination point. (U) means a unique species among the cations shown here. (Caption: element index of the combination point in *cation (a)*, element index of the combination point in *cation (b)*, bond order in between). Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.

4.1.8. Component Swap

The component swap operation achieves the generation of two novel ionic liquids

doi:10.6342/NTU202403528

through a straightforward exchange of the MDS between *anion (a)* and *anion (b)*. This process transforms *IL(a)* into [*cation(a)*][*anion(b)*] and *IL(b)* into [*cation(b)*][*anion(a)*]. It's important to note that while this operation yields distinct ionic liquids, it does not introduce new molecular species. This is because the connectivity and elemental composition of each molecule remain unaltered. Despite this, the component swap operation holds potential value in the design of diverse molecular mixtures.

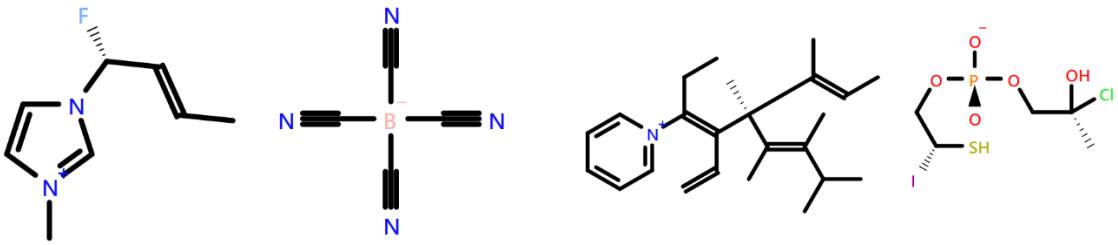


Figure 4.1-9. The *IL (a)* and *IL (b)* after *component swap* operation. Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.

4.2. Chemical Space Exploration via Iterative Enumeration

A significant strength of MARS+ lies in its capability to perform all conceivable molecular operations on every atom and bond within a given molecule. This is achieved efficiently by employing nested for-loops to iterate through all operations on all potential operation sites. Consider the addition operation as an illustrative example. The *Mol.addition(i,j,m)* function adds a base element *j* to the free valence of the *i-th* element in a molecule using a specific bond order *m*. To execute every possible addition operation, a triple loop is required to traverse all elements *i* in the molecule, all base elements *j* in

doi:10.6342/NTU202403528

the library, and all permissible bond orders m = 1 to 3. Similar loops can be readily developed for all the nine uni-molecular operations supported by MARS+.

To illustrate the power of exhaustive uni-molecular operations, we performed a five-round iteration on methane. This initial round yielded eleven unique new species: CC, CO, CN, CF, CCl, CBr, CI, CS, CP, C[PH4], and C[PH3]. Subsequently, each of these newly generated species underwent another round of exhaustive uni-molecular operations. This process was repeated five times, resulting in a total of 26,817,632 structures. The canonical SMILES strings (determined using Open Babel[65]) for these structures were then employed to identify unique species in each round, totaling 672,042 unique structures. **Figure 4.2-1** depicts the number of generated structures and unique species for each round. As evident from the figure, both quantities exhibit exponential growth with the number of exhaustive iterations.

Figure 4.2-2 visualizes the number of new structures generated by each operation. This exercise reveals several noteworthy observations. The first five operations (addition, insertion, subtraction, element change, and bond change) produce comparable numbers of structures from the second to fifth rounds of iterations. A more detailed analysis (**Figure 4.2-3**) indicates that addition and insertion are the primary operations responsible for generating new unique species. As the maximum molecular size among the design molecules increases with each iteration, the potential operation points for subtraction, element change, and bond change operations also grow concomitantly. Cis-trans and chirality inversions commence in the fourth iteration when the number of heavy atoms potentially reaches four in round three. Cyclization initiates in round five because the minimum permitted ring size is set to five. Notably, no decyclization operation was performed in the first five rounds due to the absence of ring-containing compounds in the initial four iterations.

doi:10.6342/NTU202403528

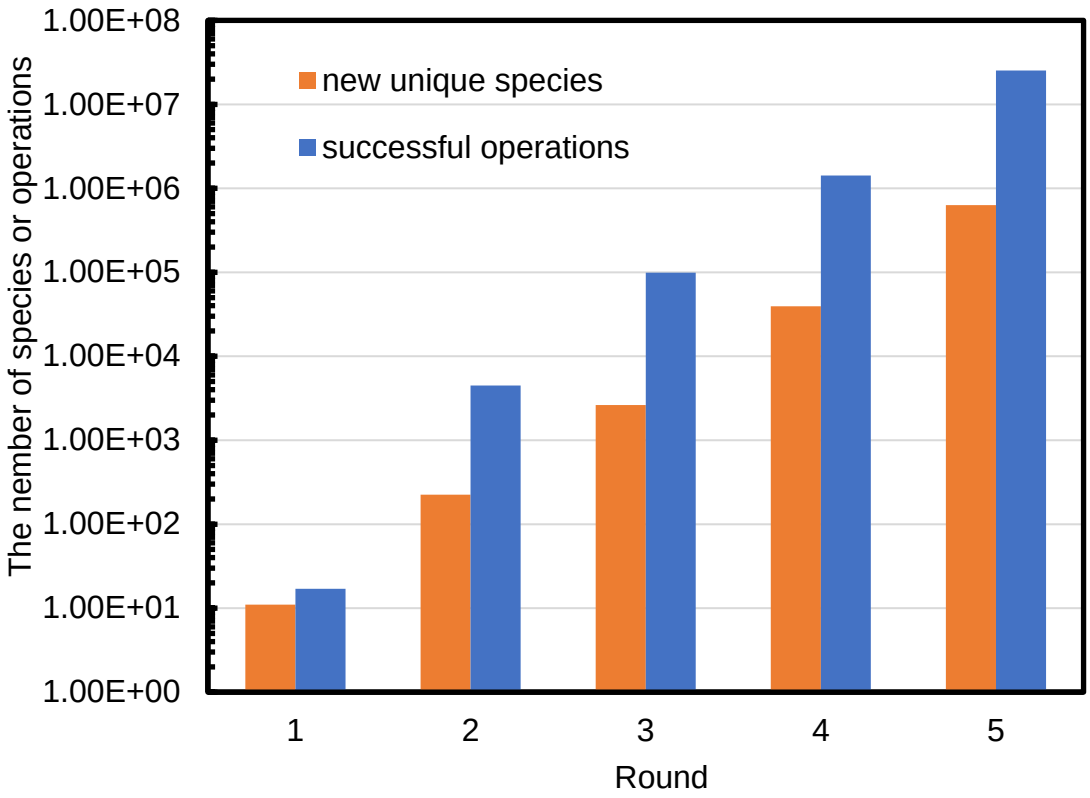


Figure 4.2-1. The number of successful operations and newly generated unique species per iteration. Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.

doi:10.6342/NTU202403528

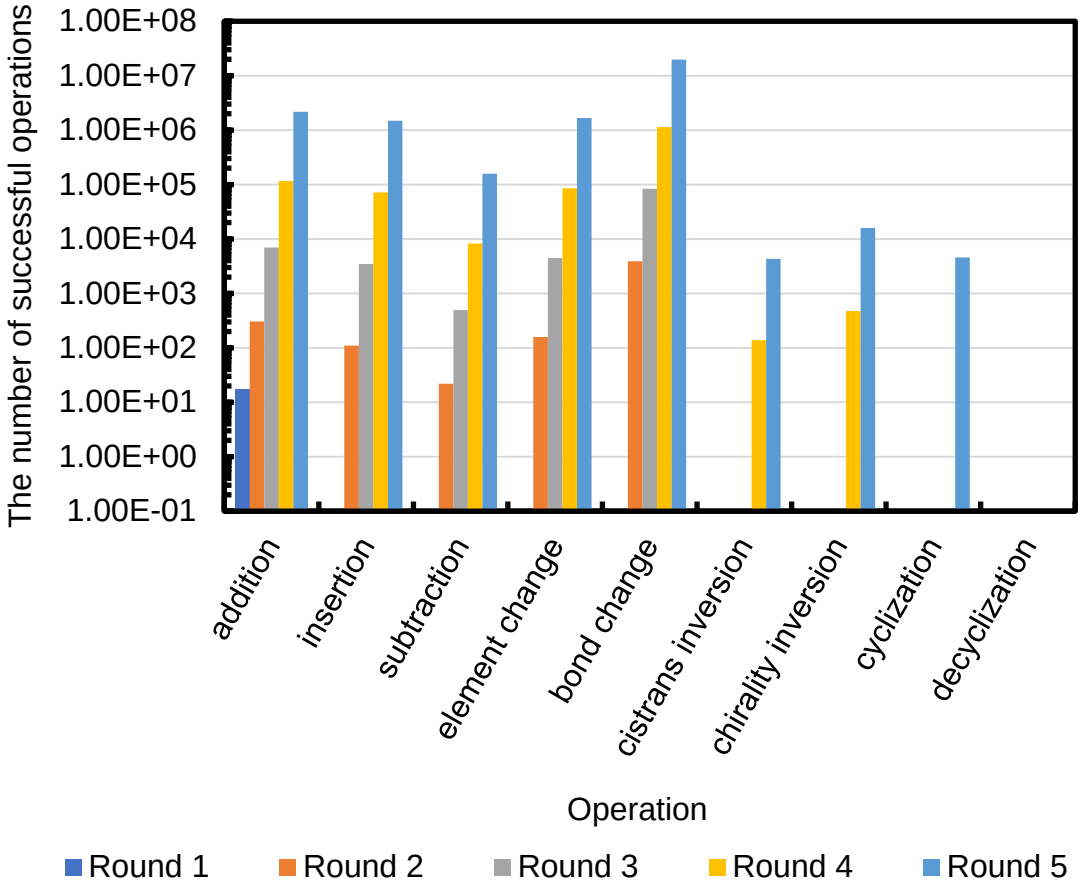


Figure 4.2-2. The number of successful operations, factorized into the contribution from each operation and each iteration. Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.

doi:10.6342/NTU202403528

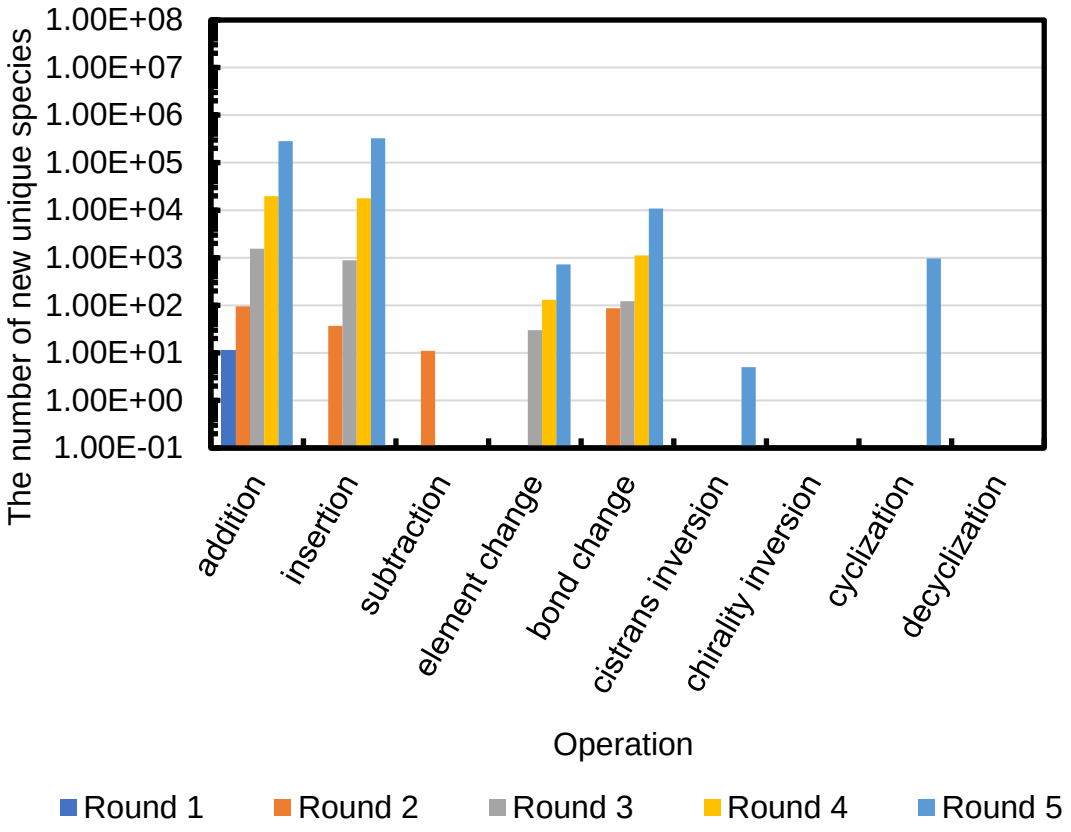


Figure 4.2-3. The number of novel unique molecules, factorized into the contribution from each operation and each iteration. Every species is credited to the operation responsible for its first appearance. Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.

4.3. Can MARS+ Produce Well-known Chemicals?

In essence, each of the molecular operations can be considered as a form of virtual elementary chemical reactions. Theoretically, there should be at least a virtual synthesis pathway between any reactant species and any product species if the operations are sufficiently "elementary". This can be preliminarily assessed by determining if there is at least a programmable sequence of molecular operations that can traverse all the species in a real synthesis pathway. If such sequences exist, the scheme of molecular operations

doi:10.6342/NTU202403528

is elementary enough to account for this synthesis pathway, and the possibility to find the involved species with computational design is justified. For this case study, we have selected the total synthesis scheme of Oseltamivir (or Tamiflu), as drug molecule syntheses are typically much more complex than ionic liquid syntheses. **Figure 4.3-1** displays the intermediate products in the synthesis pathway proposed by E.J. Corey et al.[197, 198], and the complete sequence of molecular operations we construct is shown in **Figure B1** (also see *Tamiflu_Corey()* function in *src/CASES_NEU.cpp*). There are 7 types of operations involved, including *addition*, *element change*, *bond change*, *subtraction*, *chirality inversion*, *cyclization*, and *decyclization*.

To assess the chemical feasibility of all the involved chemical structures, we use synthetic accessibility score (SAscore)[64, 168] and synthetic complexity score (SCscore)[169]. SAscore assesses the synthetic accessibility based on the occurrences of molecular fragments in PubChem database, while SCscore evaluates synthetic complexity based on the number of required reaction steps inferred from the knowledge of Reaxys database. Using these two scoring functions, we illustrate the variation in chemical feasibility against the sequential reaction steps in **Figure B2**. A practical sequential reaction should exhibit a monotonically increasing SCscore curve. Therefore, when a slump in SCscore is observed (e.g. the 10th to 11th species in **Figure B2**), it suggests that the reverse reaction may be more practical from a domain knowledge perspective. On the other hand, a surge in SAscore curve (e.g. the 16th to 19th species in **Figure B2**) indicates that rare and complex substructures, such as rings, have been formed through the reaction steps.

It should be emphasized that those reaction steps associated with reasonable variations in SAscore and SCscore curves are not necessarily practical. To obtain the most realistic reaction pathway, one may resort to computer-assisted synthesis planning (CASP)[126] software, such as AiZynthFinder[125] and ASKCOS[126]. When provided with a

doi:10.6342/NTU202403528

chemical, CASP software attempts to identify common precursors and provide practical reaction steps for synthesizing the chemical from these precursors.

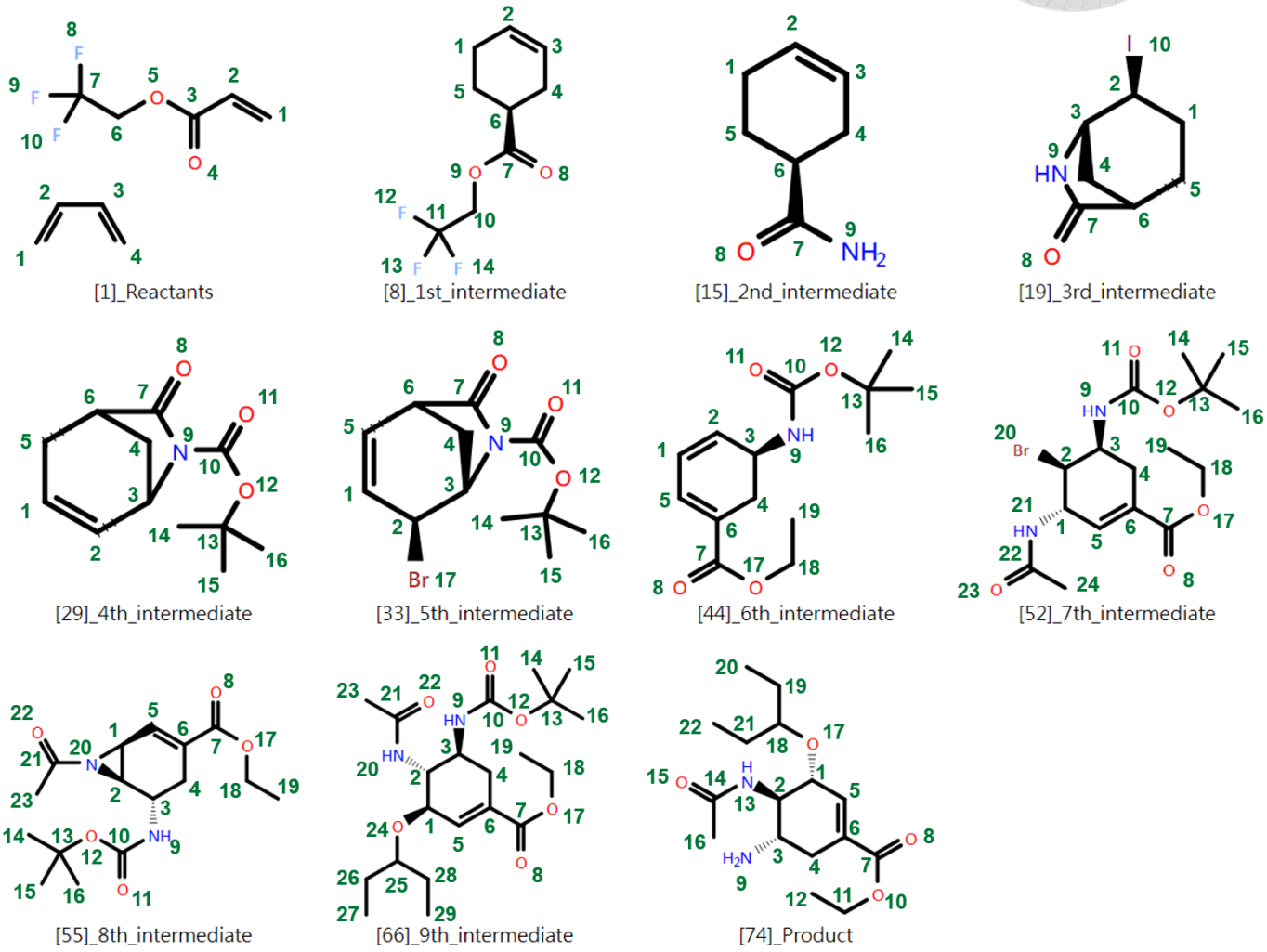


Figure 4.3-1. The intermediate products in the total synthesis scheme of Oseltamivir proposed by E.J. Corey et al.[197, 198] The green numbers are element indices. Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.

doi:10.6342/NTU202403528

Chapter 5. Design of Novel ILs for CO_2 Capture

5.1. A Review of Theoretical and Application Insights

The emission of carbon dioxide (CO_2) has emerged as a significant contributor to climate change, drawing increasing concerns in recent years.[199] Although the atmospheric CO_2 concentration surpassed the 400-ppm threshold in 2015,[200] the utilization of fossil fuels remains unavoidable in contemporary anthropogenic activities. Notably, the power generation, as well as industrial processes, are responsible for approximately two-third of 36.8 billion tons of global CO_2 emissions in 2022.[201, 202] In response to the urgent need to mitigate CO_2 emissions, various carbon capture and storage (CCS) techniques[8, 203-208] have been under development and continuous improvement, including physical absorption, chemical absorption, membrane-based separation, cryogenic distillation, chemical looping combustion, hydrate-based separation, adsorption, and so on.[205, 209]

There are essentially three schemes for the integration of these techniques into power plants and industrial processes. The pre-combustion scheme aims to separate CO_2 from fossil fuels before combustion. In this scheme, the coal (or natural gas) feed undergoes the high-temperature gasification reaction $C + H_2O \rightarrow CO + H_2$ (or steam methane reforming reaction $CH_4 + H_2O \rightarrow CO + 3\ H_2$), followed by the water-gas shift reaction $CO + H_2O \rightarrow CO_2 + H_2$ at around 40°C. [204, 210] Subsequently, carbon capture techniques are applied to the shifted syngas, leaving pure H_2 as the fuel for combustion. As shifted syngas typically contains a moderate level of CO_2 (15-60 %[8, 204, 205], $P_{CO2} \approx$ 12-20 bar[210]), physical absorption employing Selexol solvent is reportedly a suitable strategy for carbon captures in pre-combustion scheme.[210, 211] The post-combustion scheme is often implemented in coal-fired power plants, where the flue gas typically contains a low level

doi:10.6342/NTU202403528

of CO_2 (10-14 %).[8, 204, 205] Monoethanolamine (MEA)-based chemical absorption, which utilizes acid-base neutralization, is a conventional approach for post-combustion scheme. The oxyfuel-combustion scheme involves combusting fuel in an oxygen-rich environment, resulting in a flue gas mainly containing water and a high level of CO_2 (70-98 %).[8, 204, 205] After the removal of impurities (e.g. sulfur dioxide, nitrogen oxides, and fly ash), the CO_2 can be captured by compressing the flue gas.

Table 5.1-1. Typical subject gas and operating condition for carbon capture.

Schemes	**Post-combustion**[207, 212-215]	**Pre-combustion**[207, 216-221]	**Oxy-combustion**[212, 222-224]
T and P of the subject gas	1 bar, 40-75 ℃	20-60 bar, 35-40 ℃	1 bar, 150 ℃
Mol%	Flue gas	Shifted gas	Flue gas
CO_2	0.150	0.300	0.700
CO	0.000	0.050	0.000
N_2	0.650	0.000	0.150
O_2	0.100	0.000	0.050
H_2	0.000	0.450	0.000
H_2O	0.100	0.150	0.100
CH_4	0.000	0.050	0.000

Among these techniques and processes, MEA-based post-combustion scheme has the highest technological maturity,[203, 225] as evidenced by its successful commercialization in 2013[226]. However, the technique encounters challenges related to high solvent loss

doi:10.6342/NTU202403528

(volatilization, oxidative degradation, and thermal degradation), high energy demands for solvent regeneration (3.2-4.2 GJ/tonne CO_2 at >100℃), and solvent corrosivity.[208, 227, 228] The plantwide energy efficiency may be enhanced through process reconfigurations.[208, 228, 229] Nevertheless, the issues of solvent mass loss and high regeneration energy persist as challenges unless a different solvent is used, as these aspects are primarily governed by the thermodynamic nature of solvent.

In the past two decades, ionic liquids (ILs) have emerged as a potential solution to these issues.[208, 230-232] Several mechanistic studies have revealed theoretical feasibility to using them as CO_2 absorbents, although a few of their arguments are case-dependent.[233] Firstly, it has been proposed that the anion of IL serves as a Lewis base and interacts weakly with Lewis-acid CO_2. Qualitatively, the CO_2 solubility is proportional to the strength of acid-base interaction, and the basicity of the anion directly influences this interaction.[233] However, other studies also propose that it is the ether group (R_1-O-R_2)[234] and primary amine group (R-NH_2)[235] in either cation or anion part can lead to the nucleophilic reactions between ILs and CO_2. In this scenario, chemical absorption occurs, and the measured CO_2 solubility is usually significantly higher than the predicted value from physical absorption models.[236]

A suitable IL-based solvent is expected to have a weak interaction (typically van der Waals force) with CO_2, ensuring that the solvent regeneration will not be too difficult.[237] Secondly, the number of exposed binding sites per free volume is also the crucial factors.[233, 238] The molar free volume of IL is influenced by both the cation-anion interaction and the shape of the molecule. Based on different models for excluded volume, the molar free volume, $\underline{V}^f$, is often calculated from either eq (5.1–1) or eq (5.1–2).[238]

doi:10.6342/NTU202403528

$$\underline{V}^f = \underline{V} - 1.3\,\underline{V}^{vdW} \quad (5.1\text{–}1)$$

$$\underline{V}^f = \underline{V} - \underline{V}^{COSMO} \quad (5.1\text{–}2)$$

Here, $\underline{V}$ is the molar volume of IL, $\underline{V}^{vdW}$ is the molar van der Waals volume of IL, and $\underline{V}^{COSMO}$ is the molar COSMO volume of IL. Asymmetric molecular structures are generally prone to creating more free volume with particular orientation.[239] On the other hand, when the gas-solvent binding interaction is not strong, a greater free volume generally promote the gas diffusivity, allowing the solvent to accommodate more gas within a finite time interval.[240] When the gas-solvent binding interaction is strong (e.g. hydrogen bonding), it is necessary to consider that, in addition to the promoting effect mentioned above, a greater free volume may expose more binding sites of solvents, leading to a retarded diffusion.[241] The permeability of gas, $\mathcal{P}_i$, which considers these two competing effects, is suitable for evaluating solvent performance in finite-time absorption process.[240]

$$\mathcal{P}_i = \frac{D_i}{H_{i/S}} \quad (5.1\text{–}3)$$

Here, D_i represents the diffusivity of gas molecule *i*, and $H_{i/S}$ is the Henry's constant of gas molecule *i* in solution *S*. The mole fraction-based gas solubility is inversely proportional to Henry's constant.[62] In a case study on imidazolium-based ILs with various lengths of alkyl chains, a positive correlation is found between the mole fraction-based solubility and the free volume of IL.[238] Additionally, CO_2/CH_4 and CO_2/N_2 selectivity also exhibit negative correlations with molar volume of IL.[238] These evidences suggest that desirable performance may be achieved by pursuing ILs with low molar

doi:10.6342/NTU202403528

volume and high molar free volume.

ILs typically exhibit negligible vapor pressure, non-flammability, and high thermochemical stability over a wide temperature range.[242-245] These characteristics are partially ascribed to the Coulombic interaction between the cation and anion of the ILs. Furthermore, studies have indicated that the energy consumption associated with the IL solvents regeneration may be 30-50% lower than that required by conventional MEA.[246, 247] These attributes are favorable for applications in absorption processes, suggesting their potential as substitutes for MEA. The structural and compositional tunability of ILs also offer a wide spectrum of novel species and performance properties yet to be explored.[248] It is suggested that at least a million of pure ILs are theoretically possible,[242] and some of them have proven to be promising solvents for CO_2 capture applications.[230, 247] However, certain ILs also exhibit high viscosity and high molar heat capacity, leading to increased costs associated with solvent pumping and CO_2 desorption, respectively.[247] Addressing these challenges represents a key area for future research endeavors.

5.2. Thermodynamic Modeling

Considering a system of solute gas mixture and solvent at vapor-liquid phase equilibrium (VLE)[62] under temperature T and pressure P. the solubility of the solute *i* in the solvent could be determined by equilibrium criterion $\overline{f}_i^V = \overline{f}_i^L$. Here, $\overline{f}_i^V$ and $\overline{f}_i^L$ are the fugacity of component *i* in the gas phase and liquid phase, respectively. With ideal mixture (IM) chosen as the reference system for each of the two phases, the equilibrium criterion can be expressed as:

$$y_i \overline{\phi}_i(T, P, \underline{y}) P = x_i \gamma_{i/S}(T, P, \underline{x}) f_i(T, P) \qquad (5.2\text{–}1)$$

doi:10.6342/NTU202403528

Here, $\underline{y}$ and $\underline{x}$ are the equilibrium composition of the gas mixture and the liquid mixture, y_i and x_i are the equilibrium mole fraction of component *i* in the vapor phase and the liquid phase, P is the pressure of system, T is the temperature of system, $\gamma_{i/S}$ is the activity coefficient of component *i* in the solvent *S*, $\overline{\phi_i}$ is the fugacity coefficient of component *i* in the vapor phase, and f_i represents the fugacity of component *i* in the hypothetical liquid state at T and P. In the scenario of modest solubility, the infinitely dilute solute in the solvent (i.e. $x_i \to 0$) is often set as the reference state of liquid phase.

$$\overline{f}_i^L = x_i \gamma_{i/S}(T, P, \underline{x}) f_i(T, P) = x_i \gamma_{i/S}^*(T, P, \underline{x}) H_{i/S}(T, P) \tag{5.2–2}$$

Here, H_i is Henry's constant of the gas solute, $\gamma_{i/S}^*$ is the modified activity coefficient using infinite dilute solute in liquid phase as reference state. From this definition, $\gamma_{i/S}^*$ will be unity at infinite dilution ($\lim_{x_i \to 0} \gamma_{i/S}^*(T, P, \underline{x}) = 1$). Subsequently, the mathematical form for Henry's constant and a modified activity coefficient can be derived from taking the infinite dilute limit of eq (5.2–2).

Here, $H_{i/S}$ represents the Henry's constant for carbon dioxide, and $\gamma_{i/S}^*$ represents the activity coefficient defined with the reference system of carbon dioxide present as an infinitely dilute species in the ionic liquid. According to this definition,

$$\lim_{x_i \to 0} \frac{\overline{f}_i^L}{x_i} = H_{i/S}(T, P, x_i \to 0) = \gamma_{i/S}^{\infty}(T, P, x_i \to 0) f_i(T, P) \tag{5.2–3}$$

$$\gamma_{i/S}^*(T, P, \underline{x}) = \frac{\gamma_{i/S}(T, P, \underline{x})}{\gamma_{i/S}^{\infty}(T, P, x_i \to 0)} \tag{5.2–4}$$

doi:10.6342/NTU202403528

Here, $\gamma_{i/S}^{\infty}(T,P,x_i \to 0)$ is the infinite dilute activity coefficient (IDAC) of solute in solvent *S*, i.e. $\lim_{x_i \to 0} \gamma_{i/S}(T,P,\underline{x}) = \gamma_{i/S}^{\infty}(T,P,x_i \to 0)$. For region of low solubility, the activity coefficient $\gamma_{i/S}$ can be reasonably approximated by $\gamma_{i/S}^{\infty}$ (or equivalently $\gamma_{i/S}^{*}(T,P,\underline{x}) \approx 1$). This is known as the Henry's law:

$$x_i^{Henry} = \frac{y_i \overline{\phi}_i(T,P,\underline{y})P}{H_{i/S}(T,P,x_i \to 0)} = \frac{y_i \overline{\phi}_i(T,P,\underline{y})P}{\gamma_{i/S}^{\infty}(T,P,x_i \to 0) f_i(T,P)} \tag{5.2–5}$$

For region of higher solubility, Henry's law may lead to nonnegligible error due to large deviation of $\gamma_{i/S}^{*}(T,P,\underline{x})$ from unity. In this situation, one should consider the actual form of $\gamma_{i/S}^{*}(T,P,\underline{x})$ when calculating solubility. This is equivalent to return to eq (5.2–1) without further simplification. The solubility x_i should be solved through successive iterations using eq (5.2–6).

$$x_i^{VLE} = \frac{1\ bar}{\gamma_{i/S}(T,P,\underline{x}) f_i(T,P)} \tag{5.2–6}$$

In this work, the desirable equilibrium composition of the gas mixture $\underline{y}$, operating temperature T, and operating pressure P are user-specified, while the solubility $\underline{x}$ is the thermodynamic variable to be determined. From eq (5.2–6), the mathematical form of distribution coefficient (β_i), selectivity (S_{ij}), performance index (PI_{ij}), absorption-desorption index (ADI_i), and absorption-selectivity-desorption index ($ASDI_{ij}$) can be defined,[142, 249-251] with the subscript *i* and *j* denoting different chemical components. High

doi:10.6342/NTU202403528

β_i, S_{ij}, and PI_{ij} imply the potential for desirable absorption performance. When selectivity and desorption are also considered, low ADI_i and $ASDI_{ij}$ indicate the potential for desirable overall performance.

$$\beta_i = \left(\frac{x_i}{y_i}\right)_{T,P,\underline{y}} = \left(\frac{\overline{\phi}_i P}{\gamma_{i/S} f_i}\right)_{T,P,\underline{y}} \tag{5.2–7}$$

$$S_{ij} = \left(\frac{\beta_i}{\beta_j}\right)_{T,P,\underline{y}} = \left[\left(\frac{\overline{\phi}_i P}{\gamma_{i/S} f_i}\right)\left(\frac{\gamma_{j/S} f_j}{\overline{\phi}_j P}\right)\right]_{T,P,\underline{y}} \tag{5.2–8}$$

$$PI_{ij} = \left(\beta_i S_{ij}\right)_{T,P,\underline{y}} = \left[\left(\frac{\overline{\phi}_i P}{\gamma_{i/S} f_i}\right)^2\left(\frac{\gamma_{j/S} f_j}{\overline{\phi}_j P}\right)\right]_{T,P,\underline{y}} \tag{5.2–9}$$

$$ADI_i = \left[\frac{1}{(\beta_i)_{T_{ad}}}\frac{(\beta_i)_{T_{de}}}{(\beta_i)_{T_{ad}}}\right]_{P,\underline{y}} = \left[\left(\frac{\gamma_{i/S} f_i}{\overline{\phi}_j P}\right)^2_{T_{ad}}\left(\frac{\overline{\phi}_i P}{\gamma_{i/S} f_i}\right)_{T_{de}}\right]_{P,\underline{y}} \tag{5.2–10}$$

$$ASDI_{ij} = \left[\frac{1}{(\beta_i)_{T_{ad}}}\frac{1}{(S_{ij})_{T_{ad}}}\frac{(\beta_i)_{T_{de}}}{(\beta_i)_{T_{ad}}}\right]_{P,\underline{y}} = \left[\left(\frac{\gamma_{i/S} f_i}{\overline{\phi}_j P}\right)^3_{T_{ad}}\left(\frac{\overline{\phi}_j P}{\gamma_{j/S} f_j}\right)_{T_{ad}}\left(\frac{\overline{\phi}_i P}{\gamma_{i/S} f_i}\right)_{T_{de}}\right]_{P,\underline{y}} \tag{5.2–11}$$

Here, T_{ad} and T_{de} are the operating temperature of absorption and desorption processes, respectively, and the subscripts in eqs (5.2–15) ~ (5.2–11) indicate the thermodynamic condition where the physical quantities are evaluated. Since high pressure and low temperature favor the absorption of CO_2, it is most desirable to operate desorption process at low pressure and high temperature. In many studies on post-combustion CO_2 capture, the absorption performance of ILs is evaluated at 0.2 ~ 5.0 bar and 293.2 ~ 333.2 K, while the desorption performance is typically evaluated at roughly

doi:10.6342/NTU202403528

the same pressure as in the absorption process and a 30.0 ~ 90.0 K higher temperature.[231, 247, 252, 253]

In addition to solubility-based indices, the Gibbs free energy $\Delta\underline{G}_{i/S}^{abs}$, enthalpy $\Delta\underline{H}_{i/S}^{abs}$, and entropy $\Delta\underline{S}_{i/S}^{abs}$ of the standard absorption process can provide more physicochemical insights. In particular, $\Delta\underline{G}_{i/S}^{abs}$ characterizes the minimum required work to carry out the absorption process, $\Delta\underline{H}_{i/S}^{abs}$ represents the heat associated with the absorption, and $\Delta\underline{S}_{i/S}^{abs}$ roughly reflects the changed molecular ordering in solvent phase due to the absorption process. These quantities are defined as eqs (3.2–8) ~ (3.2–10).[254]

$$\Delta\overline{G}_{i/S}^{abs}(T,P,\underline{x}) = \overline{G}_{i/S}(T,P,\underline{x}) - \overline{G}_{i}^{o,IGM}(T,P^{o},\underline{x}) = RT\ln\left(\frac{\gamma_{i/S}f_{i}}{P^{o}}\right) \tag{5.2–12}$$

$$\Delta\overline{H}_{i/S}^{abs}(T,P,\underline{x}) = -T^{2}\frac{\partial}{\partial T}\left(\frac{\Delta\overline{G}_{i/S}^{abs}}{T}\right)_{P,\underline{x}} = -\frac{RT^{2}}{\gamma_{i/S}f_{i}}\left(\frac{\partial\gamma_{i/S}f_{i}}{\partial T}\right)_{P,\underline{x}} \tag{5.2–13}$$

$$\Delta S_{i/S}^{abs}(T,P,\underline{x}) = \frac{\Delta\overline{H}_{i/S}^{abs} - \Delta\overline{G}_{i/S}^{abs}}{T} = -\frac{RT}{\gamma_{i/S}f_{i}}\left(\frac{\partial\gamma_{i/S}f_{i}}{\partial T}\right)_{P,\underline{x}} - R\ln\left(\frac{\gamma_{i/S}f_{i}}{P^{o}}\right) \tag{5.2–14}$$

In this study, the activity coefficient of carbon dioxide in the ionic liquid is predicted using the COSMO-SAC (COnductor-like Screening Model - Segment Activity Coefficient) model, specifically the 2010 version[100]. The fugacity of pure liquid carbon dioxide is obtained from the NIST DIPPR 101 database using the vapor pressure equation[255], as shown in eq. (5.2–15):

$$f_{i}(T,P) \approx P_{i}^{vap}(T) = \exp\left(A_{i} + \frac{B_{i}}{T} + C_{i}\ln T + D_{i}T^{E_{i}}\right) \tag{5.2–15}$$

doi:10.6342/NTU202403528

Here, A_i, B_i, C_i, D_i, and E_i are empirical parameters, and the fugacity $f_i(T,P)$ is expressed in Pa. The fugacity of pure liquid carbon dioxide at 298.15 K is 64.48 bar.

5.3. Validation of COSMO-SAC Predictions

The reliability of property prediction model is crucial in a CAMD task, as it directly impacts the credibility of the reported optimal species. To assess the accuracy of the COSMO-SAC-based method, predicted CO_2 solubility in various IL systems is compared with experimental data. A total of 620 Henry's constant data points (105 IL species) and 4537 VLE solubility data points (96 IL species) are sourced from the ILThermo database.[256, 257] However, chemical absorption is reported to occur in some CO2-IL systems under experimental conditions. For example, ether group (R_1-O-R_2)[234] and primary amine group (R-NH_2) may react with CO_2 through nucleophilic reactions.[235] In chemical absorption, it is noted that vapor pressure is significantly lower than estimated by physical absorption models[258], indicating that CO_2 solubility in IL due to chemical absorption can be substantially higher than that from physical absorption[236] Notably, since the COSMO-SAC-2010 model does not account for chemical reactions between solute and solvent, such data points are excluded from this validation study.

These chemical absorbents include [C2mim][Ac][258], [C4mim][Ac][258], [C6mim][eFAP][258], [C4mim][PRO][258], [C4mim][ISB][258], [C4mim][Me3Ac][258], [C4mim][LEV][258], [N0,0,0,2-OH][Ac][259], [N0,0,0,2-OH][LAC][259], [(COC)mim][TFLA][234], [(COC)mim][TF2N][234], [(COC)mim][DCA][234], [(COC)mim][PF6][234], and [(COC)mim][BF4][234] belong to this type. The experiments were conducted by Sharma et al. (T=303.15~323.15 K, P=0.1~1.6 bar)[234], Yokozeki et al. (at T=298.15 K, P=0~20 bar)[258], and Kurnia et al. (T=298.15~328.15 K, P=1.16~15.56 bar)[259].

doi:10.6342/NTU202403528

Based on the common operational conditions for pre-combustion scheme (**Table 5.1-1**), data points with temperatures exceeding 350 K or pressures exceeding 60 bar are filtered out from both the VLE and Henry's constant datasets. This filtering leaves 3004 VLE solubility data points and 546 Henry's constant data points, serving as the validation sets for COSMO-SAC prediction at post-combustion or pre-combustion operating condition. Additionally, VLE data points with pressures below 5 bar are collected separately to serve as another validation set for COSMO-SAC prediction at post-combustion operating condition.

For these systems, we calculated the errors between COSMO-SAC predictions and experimental values, including the average absolute deviation (AAD), average absolute relative deviation (AARD), and the root mean square deviation (RMSD). Taking solubility as an example, the three deviations are calculated by eq. (5.3–1) to (5.3–3) respectively, and the results are presented in **Table 5.3-1**.

$$\mathrm{AAD} = \frac{1}{N_s}\sum_{i}^{N_s}\frac{1}{N_{p,i}}\sum_{j}^{N_{p,i}}\left|x_{CO_2,calc,j} - x_{CO_2,expt,j}\right| \times 100\ \% \tag{5.3–1}$$

$$\mathrm{AARD} = \frac{1}{N_s}\sum_{i}^{N_s}\frac{1}{N_{p,i}}\sum_{j}^{N_{p,i}}\frac{\left|x_{CO_2,calc,j} - x_{CO_2,expt,j}\right|}{x_{CO_2,expt,j}} \times 100\ \% \tag{5.3–2}$$

$$\mathrm{RMSD} = \sqrt{\frac{1}{N_s}\sum_{i}^{N_s}\frac{1}{N_{p,i}}\sum_{j}^{N_{p,i}}\left(x_{CO_2,calc,j} - x_{CO_2,expt,j}\right)^2} \tag{5.3–3}$$

Here, N_s represents the number of ionic liquid species, and $N_{p,i}$ represents the number of data points for ionic liquid species *i*. Furthermore, by comparing the scatter plots (**Figure 5.3-1** and **Figure 5.3-3**) with experimental data, we believe that COSMO-SAC shows acceptable performance in terms of the errors in both solubility cases,

doi:10.6342/NTU202403528

although there is still room for improvement. It can be observed that the trends in predicted values by the COSMO-SAC model are consistent with the experimental values. Another study[236], which evaluates the accuracy of COSMO-RS in predicting the CO_2 Henry's constant within a distinct set of ILs not covered in this research, reports an AARD of 64.5% for physical IL absorbents, which decreases to 29.4% after calibration. Therefore, we posit that COSMO-SAC can serve as a qualitative or semi-quantitative predictive tool within the operational range (i.e. P = 1 bar, T = 298.15 to 348.15 K) of this study.

Table 5.3-1. The accuracy of COSMO-SAC prediction: Henry's constant of CO_2 in ILs.

All the Henry's constant experimental data: 105 IL species, 620 data points			
Property	AAD	AARD	RMSD
H_{CO_2}	52.427 bar	11088 %	144.76 bar
$x_{CO_2}^{Henry}$ at 1 bar†	0.0632	46.61 %	0.2346
Subset 1 (T ≤ 350 K): 96 IL species, 546 data points			
Property	AAD	AARD	RMSD
H_{CO_2}	41.393 bar	53.836 %	129.82 bar
$x_{CO_2}^{Henry}$ at 1 bar†	0.00791	42.848 %	0.0137

† Based on eq (5.2–5), $x_{CO_2.expt}^{Henry} = 1/H_{CO_2.expt}$ at 1 bar.

doi:10.6342/NTU202403528

Table 5.3-2. The accuracy of COSMO-SAC prediction: CO_2 solubility in ILs.

All the VLE experimental data: 96 IL species, 4537 points			
Property	AAD	AARD	RMSD
P	1.248 bar	739.62 %	2.0900 bar
$x_{CO_2}^{VLE}$	0.0977	52.52 %	0.1515
Subset 1: P ≤ 60 bar, T ≤ 350 K, 80 IL species, 3004 points			
Property	AAD	AARD	RMSD
P	0.820 bar	61.28 %	1.349 bar
$x_{CO_2}^{VLE}$	0.0547	40.83 %	0.0900
Subset 2: P ≤ 5 bar, T ≤ 350 K, 50 IL species, 612 points			
Property	AAD	AARD	RMSD
P	0.931 bar	55.20 %	1.280 bar
$x_{CO_2}^{VLE}$	0.01392	58.60 %	0.0246

doi:10.6342/NTU202403528

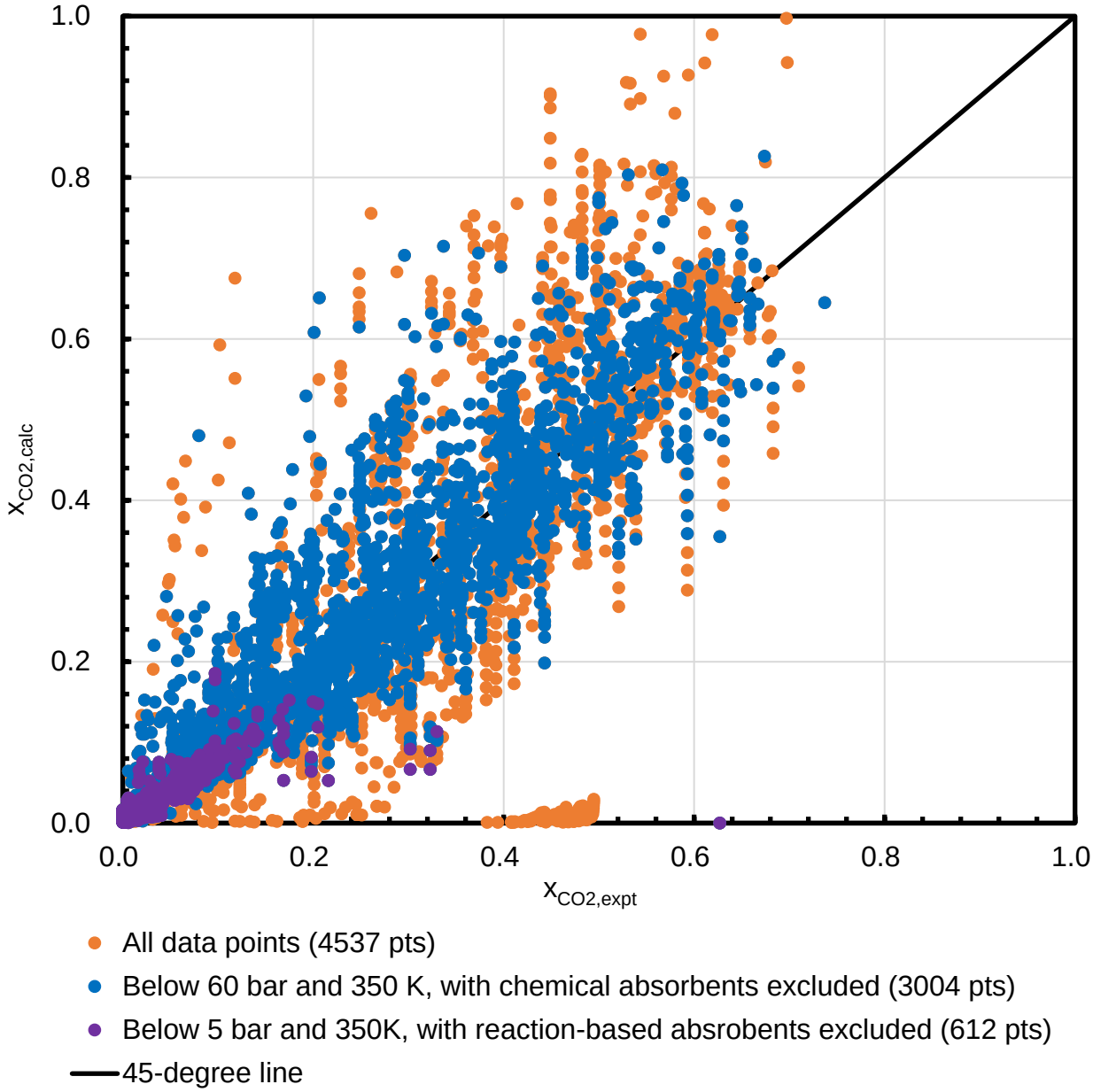


Figure 5.3-1. Comparison of COSMO-SAC predicted CO_2 solubility in ionic liquids (ILs) with VLE experimental data.

doi:10.6342/NTU202403528

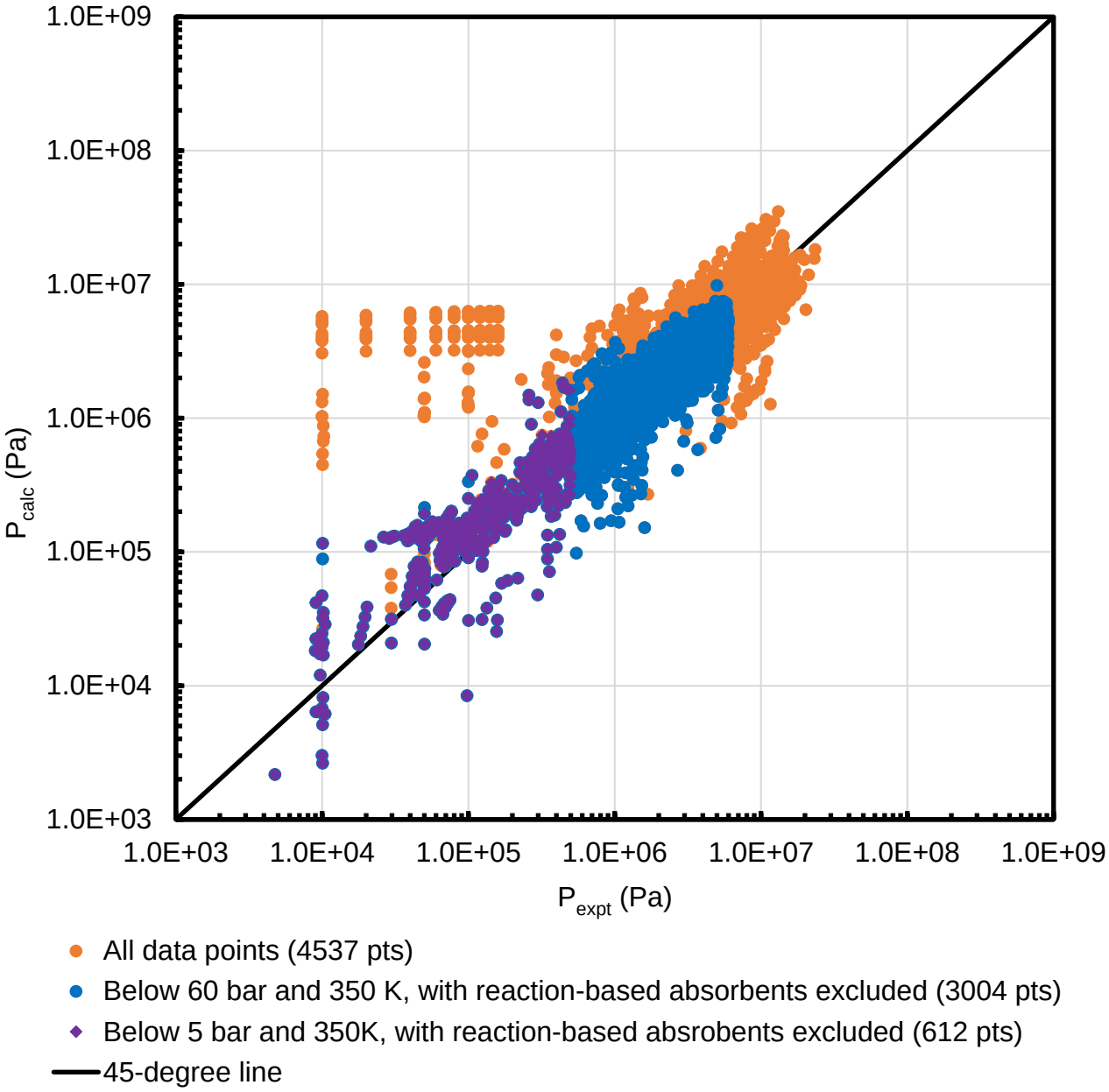


Figure 5.3-2. Comparison of COSMO-SAC predicted Henry's constant of CO_2 in ionic liquids (ILs) with experimental data.

doi:10.6342/NTU202403528

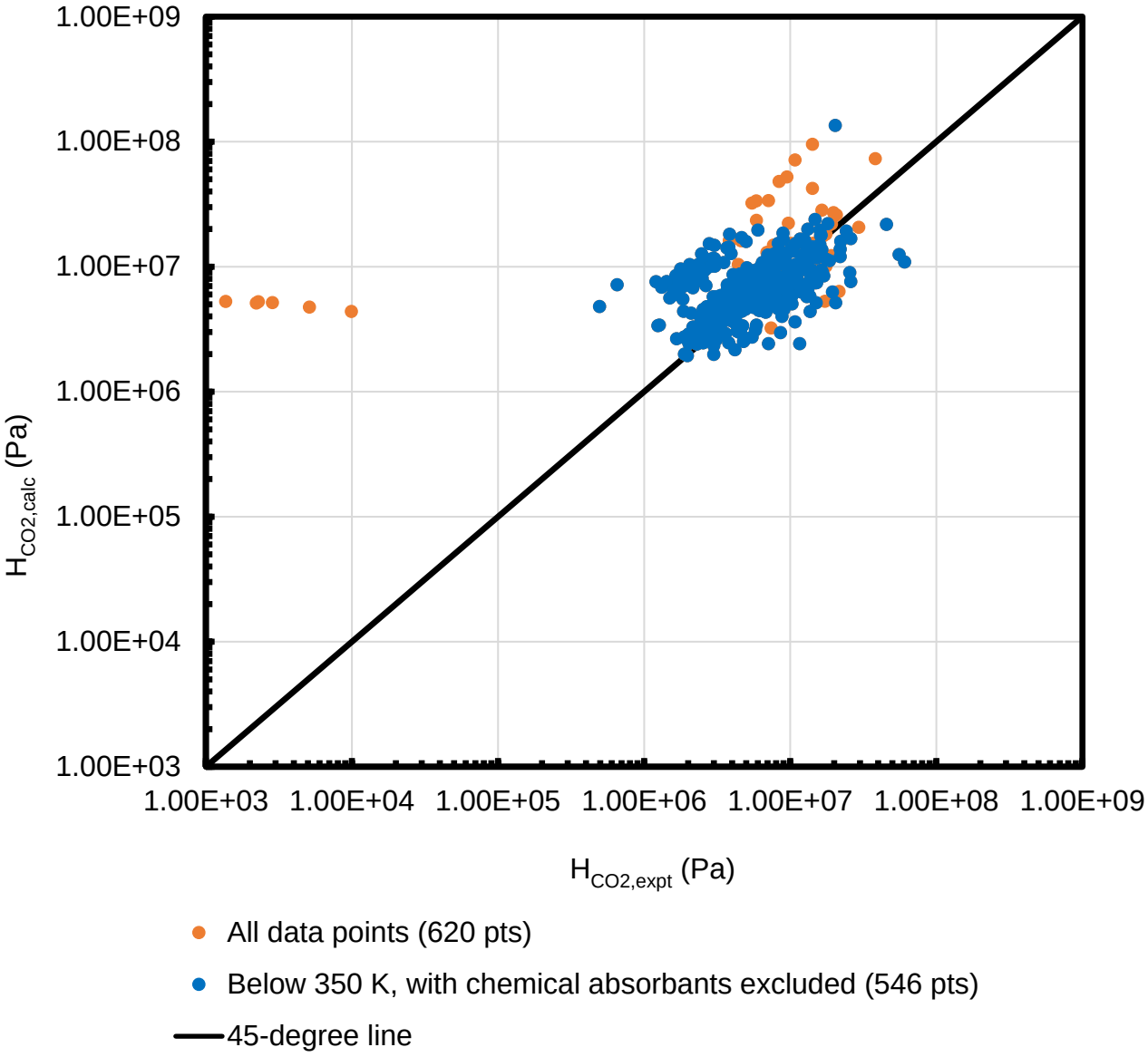


Figure 5.3-3. Comparison of COSMO-SAC predicted Henry's constant of CO_2 in ionic liquids (ILs) with experimental data.

5.4. IL Screening Using Experimentally Validated Ions

Section 5.3 presents experimental data for CO_2 capture in various ionic liquids (ILs) under diverse conditions. To facilitate a more standardized comparison, COSMO-SAC simulations are performed to evaluate their performance at a common point: T = 298.15 K and P = 1 bar. This in silico approach also allowed us to explore the full cation-anion

doi:10.6342/NTU202403528

combinations represented by the experimental data. Identifying potentially superior ILs within this unexplored space becomes a possibility. Essentially, this combined approach functions as a computational component-screening method for designing novel ILs.

Figure 5.4-1 depicts a heatmap illustrating the predicted CO_2 solubility across all screened ILs. ILs with at least an existing experimental data point are marked with dots. Notably, the heatmap reveals a combinatorically optimal IL that appears to have been missed by the experimental studies. These potentially promising ILs are listed in **Table 5.4-1**

Table 5.4-1. Some potentially promising ILs discovered from screening method.

Abbreviation	SMILES	x_{CO2}^{VLE}
[C2TT][Cl]	CCSC(=[N+](C)C)N(C)C.[Cl-]	0.0704
[C2TT][Br]	CCSC(=[N+](C)C)N(C)C.[Br-]	0.0625
[P6,6,6,14][IDA]	CCCCCCCCCCCCCC[P+](CCCCCC)(CCCCCC)CCCCCC.[O-]C(=O)CNCC(=O)[O-]	0.0669
[C2TT][IDA]	CCSC(=[N+](C)C)N(C)C.[O-]C(=O)CNCC(=O)[O-]	0.0626
[C1mim][Cl]	Cn1cc[n+](c1)C.[Cl-]	0.0589

doi:10.6342/NTU202403528

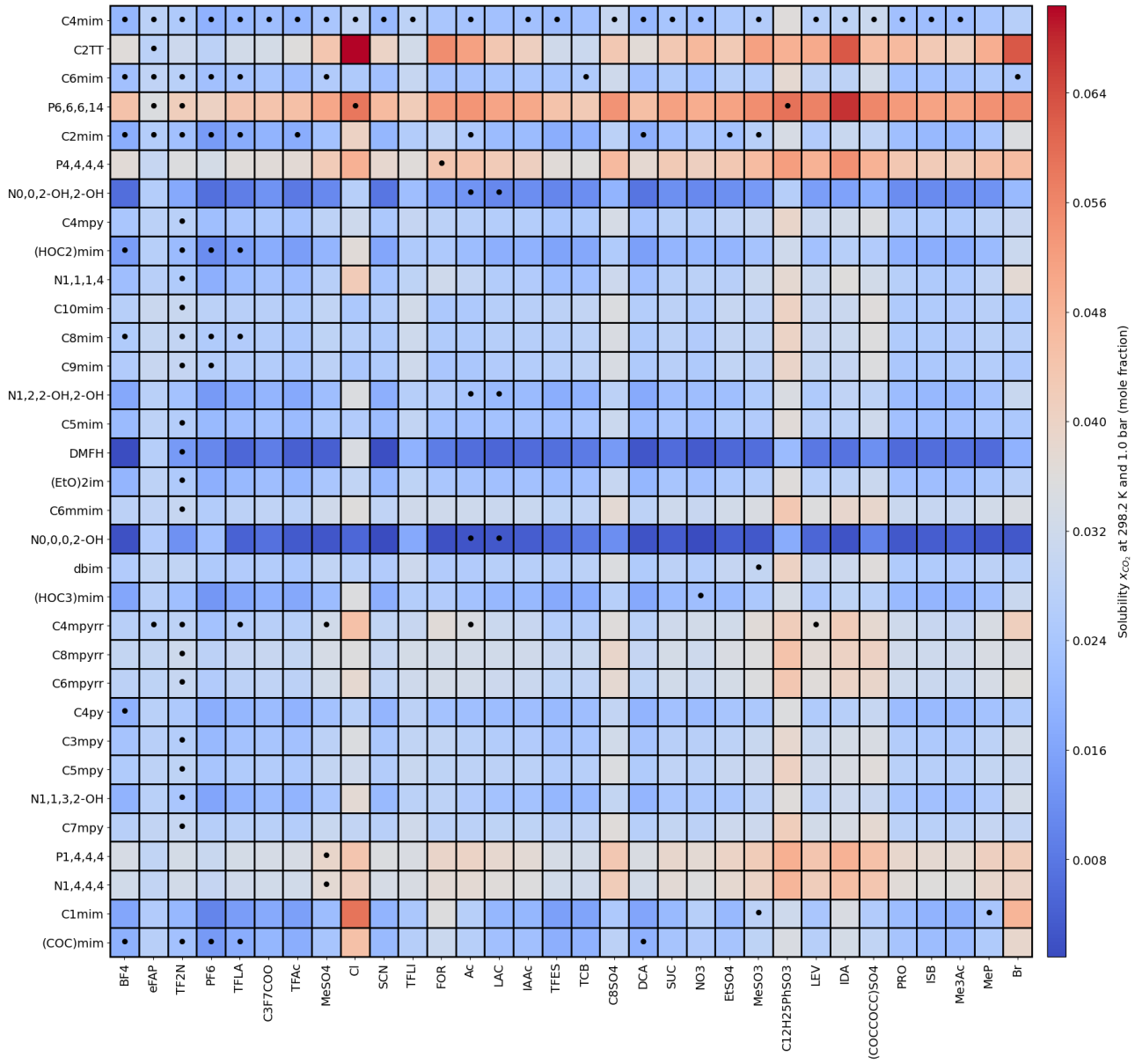


Figure 5.4-1. Solubility of CO_2 in screened ILs. A dot in a cell indicates the presence of at least one VLE experimental data point for the CO_2-IL system.

Figure 5.4-2 and **Figure 5.4-3** demonstrate the heatmaps of reciprocal SAscore and reciprocal SCscore, respectively, for every screened ILs. As mentioned in section 3.2.3, a chemical with its SAscore larger than 4.0 (or reciprocal SAscore $\leq$ 0.25) is considered a rare molecular structure. Based on this criterion, these results suggest that a practical IL may not always be assessed as highly feasible according to these two indices. Note that

doi:10.6342/NTU202403528

the SAscores for the screened ILs range from 2.272 ([P6,6,6,14][Cl]) to 5.79 ([N0,0,2-OH,2-OH][PF6]).

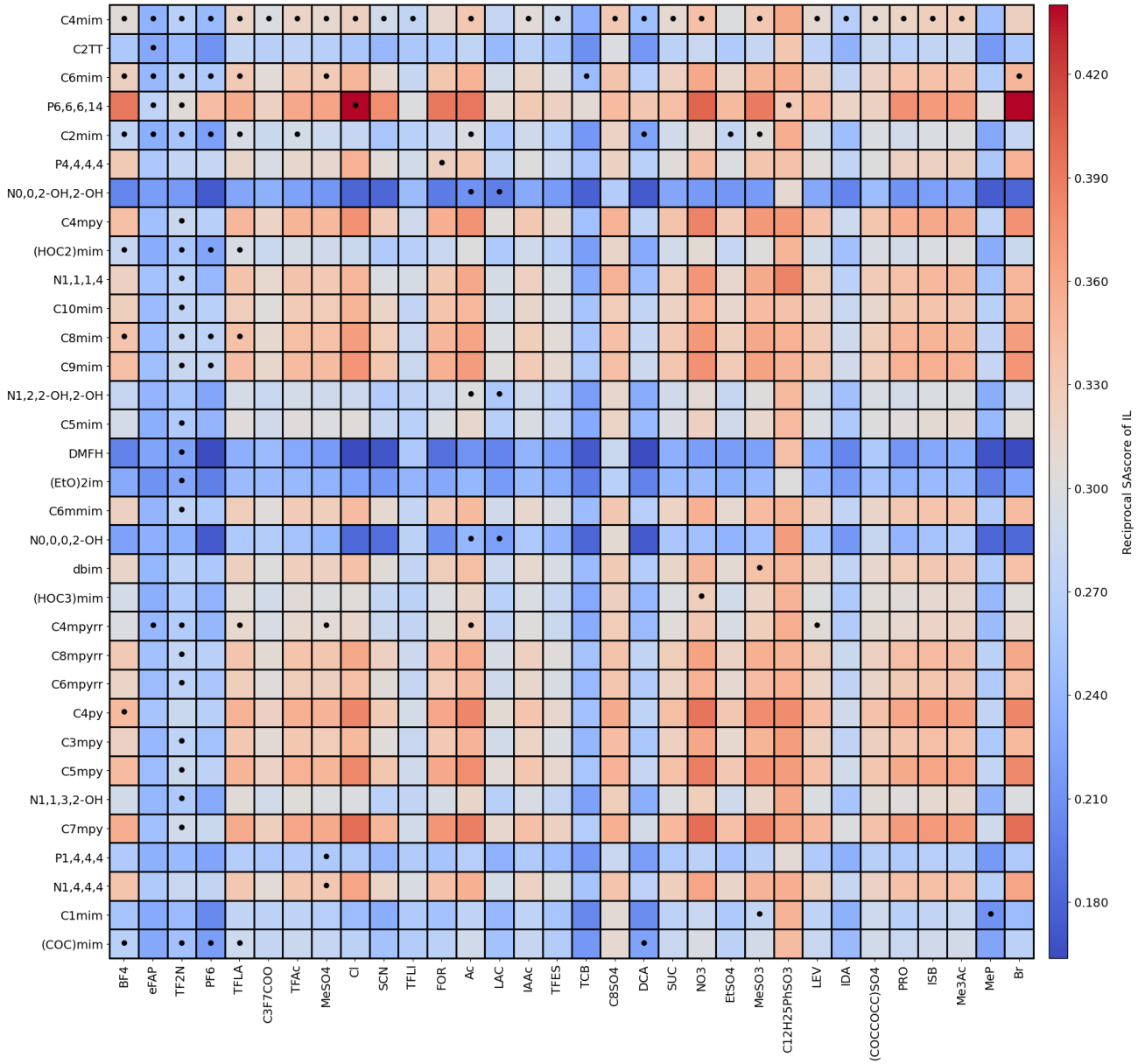


Figure 5.4-2. Reciprocal SAscore of screened ILs. Red cells indicate high synthetic accessibility (or low structural complexity). A dot in a cell indicates the presence of at least one VLE experimental data point for the CO_2-IL system.

doi:10.6342/NTU202403528

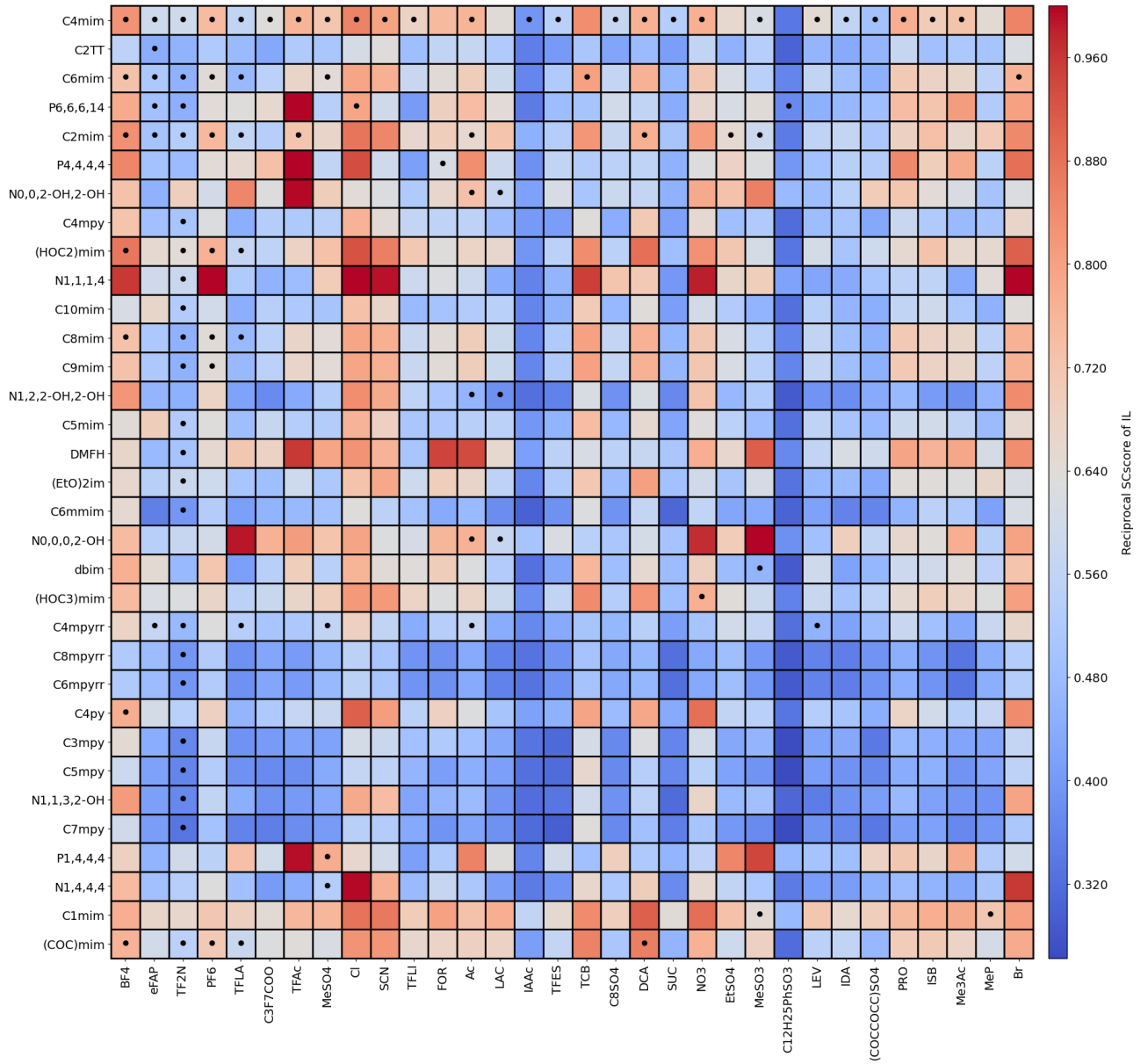


Figure 5.4-3. Reciprocal SCscore of screened ILs. Red cells indicate low synthetic complexity. A dot in a cell indicates the presence of at least one VLE experimental data point for the CO_2-IL system

The absorption free energy ($\Delta\overline{G}_{i/S}^{abs}$), absorption enthalpy ($\Delta\overline{H}_{i/S}^{abs}$), and absorption entropy ($\Delta\overline{S}_{i/S}^{abs}$) of the screened ILs are presented in **Figure 5.4-4**, **Figure 5.4-5**, and **Figure 5.4-6**. These represent the thermodynamic changes associated with the absorption

doi:10.6342/NTU202403528

of ideal-gas CO_2 (balanced by inert gas) at standard pressure $P^o = 1\ bar$. A positive value for Gibbs free energy indicates that the CO_2 capture process will not occur spontaneously, necessitating thermodynamic work to initiate the process. This is due to the high vapor pressure of CO_2 at 298.15 K (64.48 bar), where CO_2 tends to favor the vapor phase unless strongly interacted with by the IL, making it significantly non-ideal. The negative enthalpy indicates an exothermic absorption process, albeit typically less exothermic than conventional monoethanolamine (MEA)-based absorption processes.[255, 260] Absorption entropy characterizes the increase in the number of accessible states after the absorption process. It is intuitive to expect that CO_2 loses some configurational states upon entering a liquid phase where the solvent exhibits attractive interactions.

doi:10.6342/NTU202403528

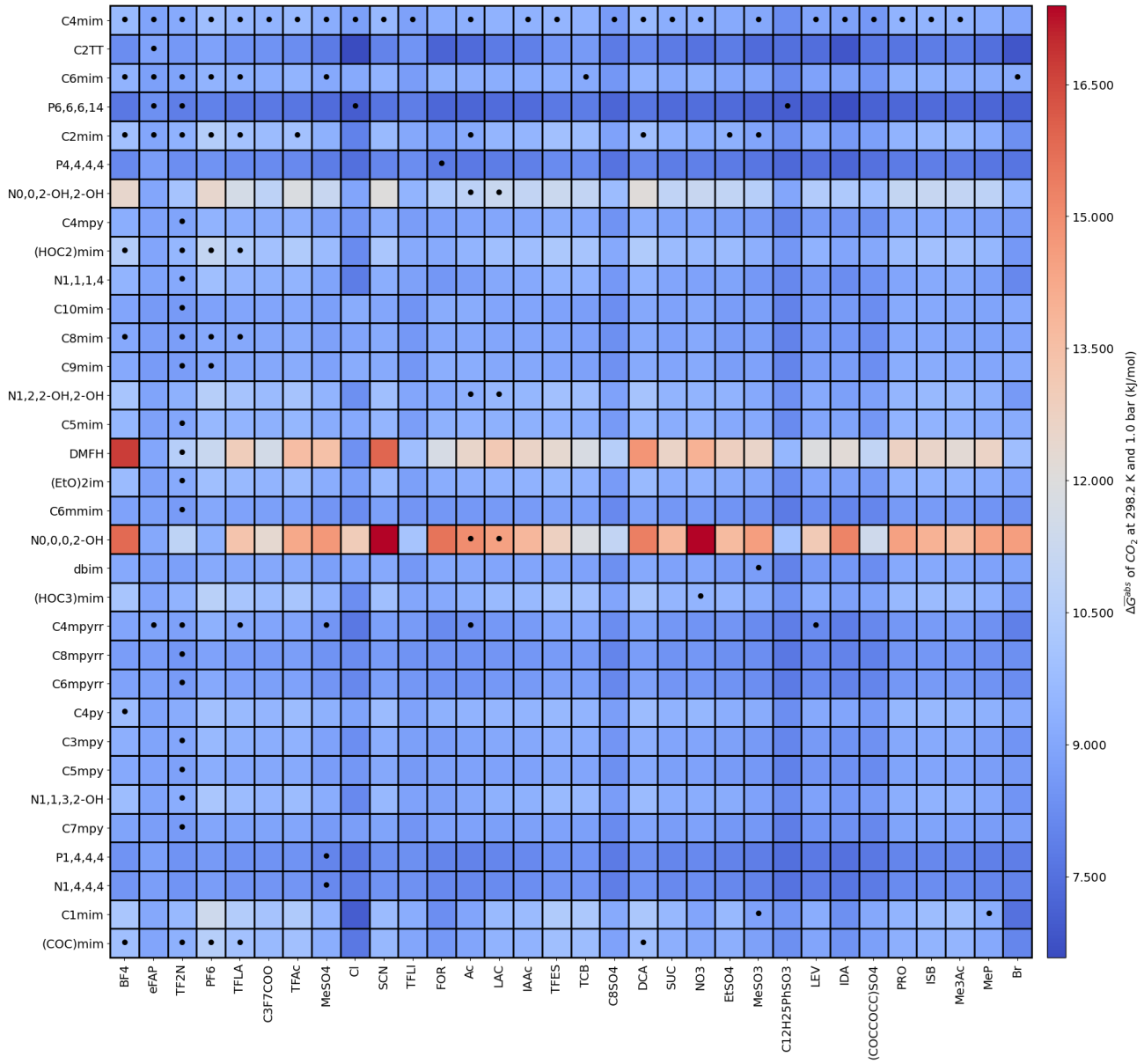


Figure 5.4-4. Absorption free energy ($\Delta\overline{G}_{i/S}^{abs}$) of CO_2 in the screened ILs. A dot in a cell indicates the presence of at least one VLE experimental data point for the CO_2-IL system.

doi:10.6342/NTU202403528

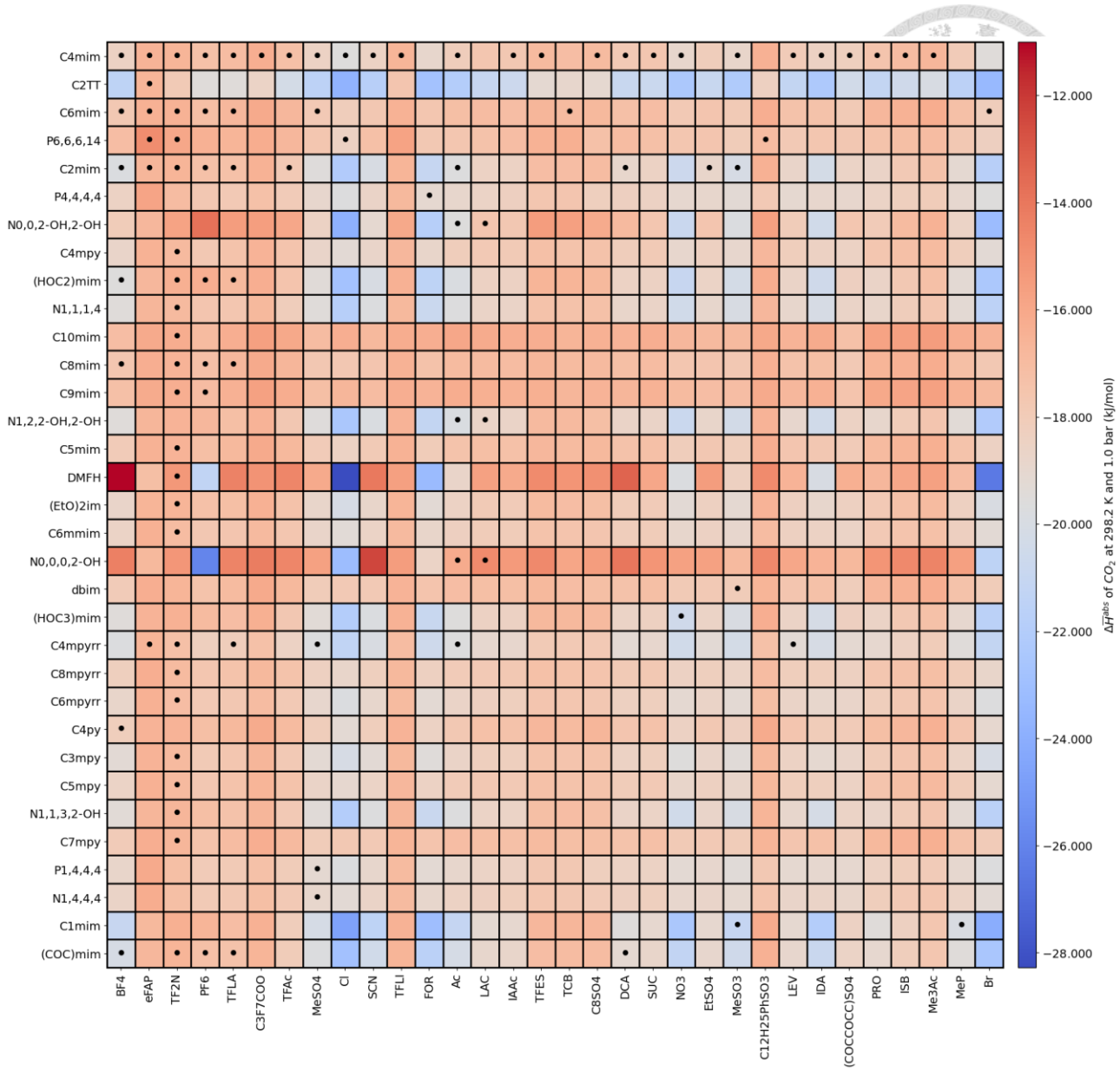


Figure 5.4-5. Absorption enthalpy ($\Delta\overline{H}_{i/S}^{abs}$) of CO_2 in the screened ILs. A dot in a cell indicates the presence of at least one VLE experimental data point for the CO_2-IL system.

doi:10.6342/NTU202403528

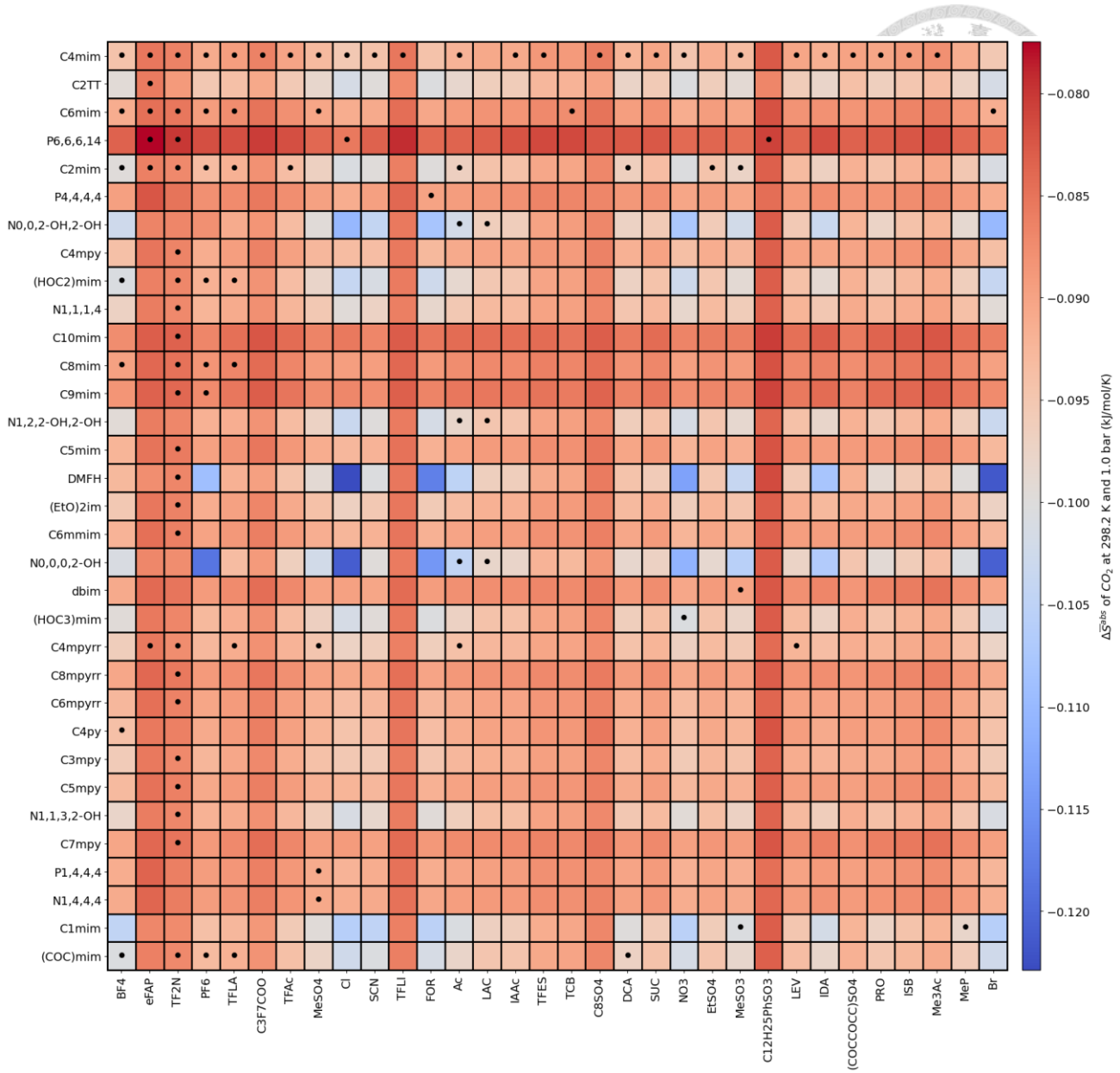


Figure 5.4-6. Absorption enthalpy ($\Delta\overline{S}_{i/S}^{abs}$) of CO_2 in the screened ILs. A dot in a cell indicates the presence of at least one VLE experimental data point for the CO_2-IL system.

Finally, the reciprocal of absorption-desorption index (ADI^{-1}, see eq (5.2–10)) represents an overall ease of employing an ionic liquid (IL) for CO_2 capture. Ideally, an IL should readily absorb CO_2 at a lower temperature and then release it at a higher temperature for regeneration. In this study, we evaluated ADI using absorption and desorption temperatures of 298.15 K and 348.15 K, respectively. Interestingly, the

doi:10.6342/NTU202403528

optimal species identified in CO_2 solubility calculations (**Figure 5.4-1**) coincide with those found in this analysis. These species exhibit relative negative absorption enthalpy (**Figure 5.4-5**). This suggests that the regeneration process, where CO_2 is desorbed from the IL, might require additional energy input due to the exothermic nature of the absorption process.

It is noteworthy that these optimal species also perform optimally among the screened ILs in terms of absorption-selectivity-desorption indices (ASDI), exception for the case that H_2O is present. Please refer to **Figure B3** to **Figure B7** for further details. Also see **Figure B8** to **Figure B13** for the IL's selectivity of CO_2 over other gas.

doi:10.6342/NTU202403528

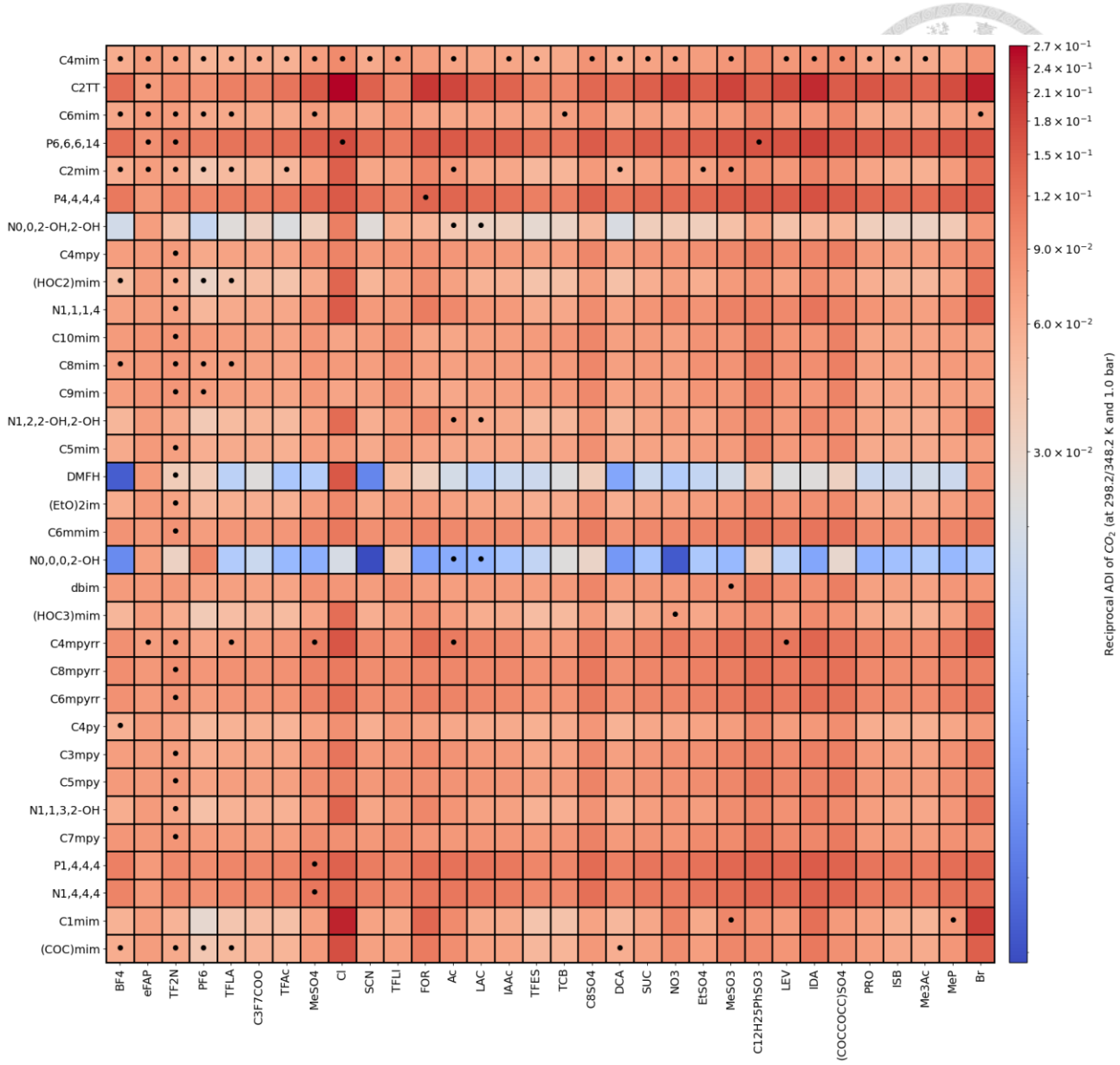


Figure 5.4-7. Reciprocal absorption-desorption index (ADI) of CO_2 in the screened ILs. The darker red indicates the desirable performance. A dot in a cell indicates the presence of at least one VLE experimental data point for the CO_2-IL system.

5.5. Computational Details of IL Design Using CAMD

Figure 5.5-1 shows the detailed steps for the design of new ionic liquids with desirable CO_2 solubility. In this work, the target values of both Henry's constant and VLE-based CO_2 solubility are set as 1.0, while the molecular size for cation and anion are not

doi:10.6342/NTU202403528

restricted. Before starting a CAMD, a library of basic elements and associated occurrence rate is prepared in MDS (section 3.1.1), and the generation of new species from genetic operations and random constructions is subjected to this probability distribution.

The CAMD process initializes with a population of 40 ILs (in MDS format) created either by user specification or by random combination of basic elements. Each IL in population can be outputted as the simplified molecular-input line-entry system (SMILES)[261] format by *mds2smi()* subroutine in MARS-PLUS package, and subsequently the SMILES is converted to 3D molecular structure with the aid of open-source program OpenBabel.[81] The 3D molecular structure is one of the proper input format for quantum chemical calculations using Gaussian 09[262]. For each molecular or ionic species, a molecular geometry optimizations in vacuum is performed on Gaussian, followed by the COSMO solvation calculation in water solvent, with both of the steps are at b3lyp/6-31g(d,p) level. After COSMO calculations, the activity, VLE-based solubility, and Henry's constant of CO_2 in an IL solvent can be determined (see section 5.2). Subsequently, the fitness $Fitfcn(\boldsymbol{u_i}, \boldsymbol{w_i}; \boldsymbol{t})$ of each IL is determined based on eq. (3.4–1) and the survival probability $P^{RW}(\boldsymbol{m_i}, \boldsymbol{s_i}; \boldsymbol{t})$ is calculated using eq. (3.4–4). Based on the probability distribution over all the species in the initial population, 40 ILs are selected using roulette wheel selection. It should be noted that the species composition of the selected ILs are usually different from that of initial population.

Some of the selected ILs species are modified into other species by applying genetic operators (section 3.3) to them. In this work, each operation is devised to manipulate a particular fraction of the selected species, namely, $(P_{cr}, P_{mu}, P_{cb}, P_{cs}) = (0.8, 0.3, 0.15, 0.15)$ for crossover, mutation, combination, and component swap, respectively. After the property evaluations for newly generated species, a new generation of population is then formally formed. The new population is subjected to next round of

doi:10.6342/NTU202403528

selection, genetic operations, and property evaluations, until the convergence or termination criteria are satisfied. However, genetic algorithm cannot offer convergence guarantees due to the stochastic nature.[263] Therefore, we set the 625th generation as the termination criterion.

To prevent the optimization from being trapped in local extrema, the alienization of population is carried out in population every 25 generations. In this operation, the ILs with their fitness lower than the 34th percentile in the population will be replaced with randomly generated chemicals. This help the optimizer discard the less-promising temporary solutions and direct the search to different locations in the feasible region. The redistribution of solutions in chemical space is found to be useful for practical design tasks.[264]

doi:10.6342/NTU202403528

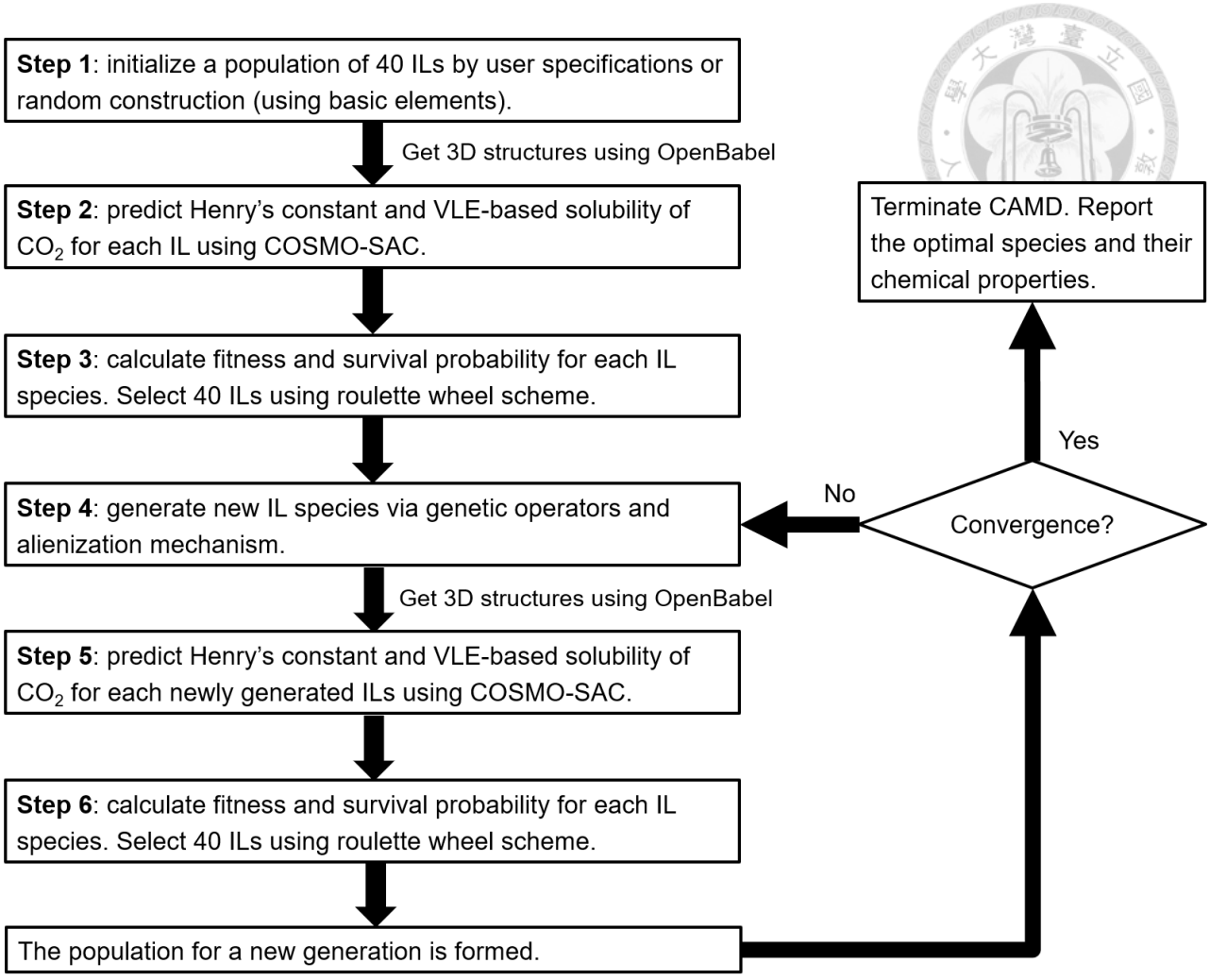


Figure 5.5-1. Flow diagram of the CAMD algorithm developed in this work.

The effect of the initial population on the result of CAMD is examined by comparing two cases of CAMD**.** In one case, 40 ILs with the best CO_2 solubility ($0.0165 \leq x_{CO_2} \leq 0.0402$) are selected from the Henry's constant data and specified as initial population. In the other case, the initial population are generated from random connections of basic elements.

doi:10.6342/NTU202403528

Table 5.5-1. Summary of the settings for the two CAMD tasks.

	Task 1	Task 2
Target Henry's constant (bar)	1.00	1.00
Target VLE-based solubility	1.00	1.00
Population	40	40
Restriction of molecular size	None	None
Initial population	40 ILs from the experimental data of Henry's constant	Randomly generated 40 ILs
Maximum of GA generations	625	625

5.6. CAMD Results

Figure 5.6-1 displays the evolution trajectory of CO_2 solubility, represented by several statistical quantities for the population, including maximum, minimum, mean value, and quartiles. For the iterations at which the alienization operator is not activated, the variation of these statistical quantities with respect to the generation are usually small (less than 0.03) in task 1. In contrast, the temporal variation observed in task 2 is relatively larger than in task 1. Specifically, in task 2, the distribution of CO_2 solubility across the population appears more scattered compared to task 1. The stabilizing effect observed in task 1 may be attributed to the similarity among the cations and anions present in the population.

doi:10.6342/NTU202403528

It is noteworthy that, in both tasks, ILs with high CO_2 solubility are primarily generated through the alienization operator. This suggests that relying solely on crossover and mutation operations may make it challenging to discover ILs with significantly enhanced CO_2 solubility. Alienization operator can mitigate the issue of CAMD becoming trapped in local maxima, where an optimal species dominates the population. By introducing new "genes" into the population, the alienization operator facilitates the exchange of molecular fragments with heterogeneous species, potentially leading to a wider variety of optimal candidates. Nevertheless, in task 2, the alienization operator appears less effective in achieving substantial solubility improvements; for instance, few ILs achieve solubility greater than 0.2 upon alienization."

Figure 5.6-2 illustrates the evolution of the number of cations, anions, and IL species within the population. The trajectory indicates that the genetic algorithm does not exhibit clear convergence under our parameter settings for both tasks. Sharp peaks in the trajectory correspond to instances of the alienization operator. Specifically, in task 1, the alienization operator effectively renews the entire population around the 175th generation, suggesting that an optimal species dominates approximately two-thirds of the population before alienization within that generation interval. Furthermore, the variety of anions generally appears lower compared to that of cations. This suggests that reducing the degree of freedom for anions contributes to achieving higher CO_2 solubility.

doi:10.6342/NTU202403528

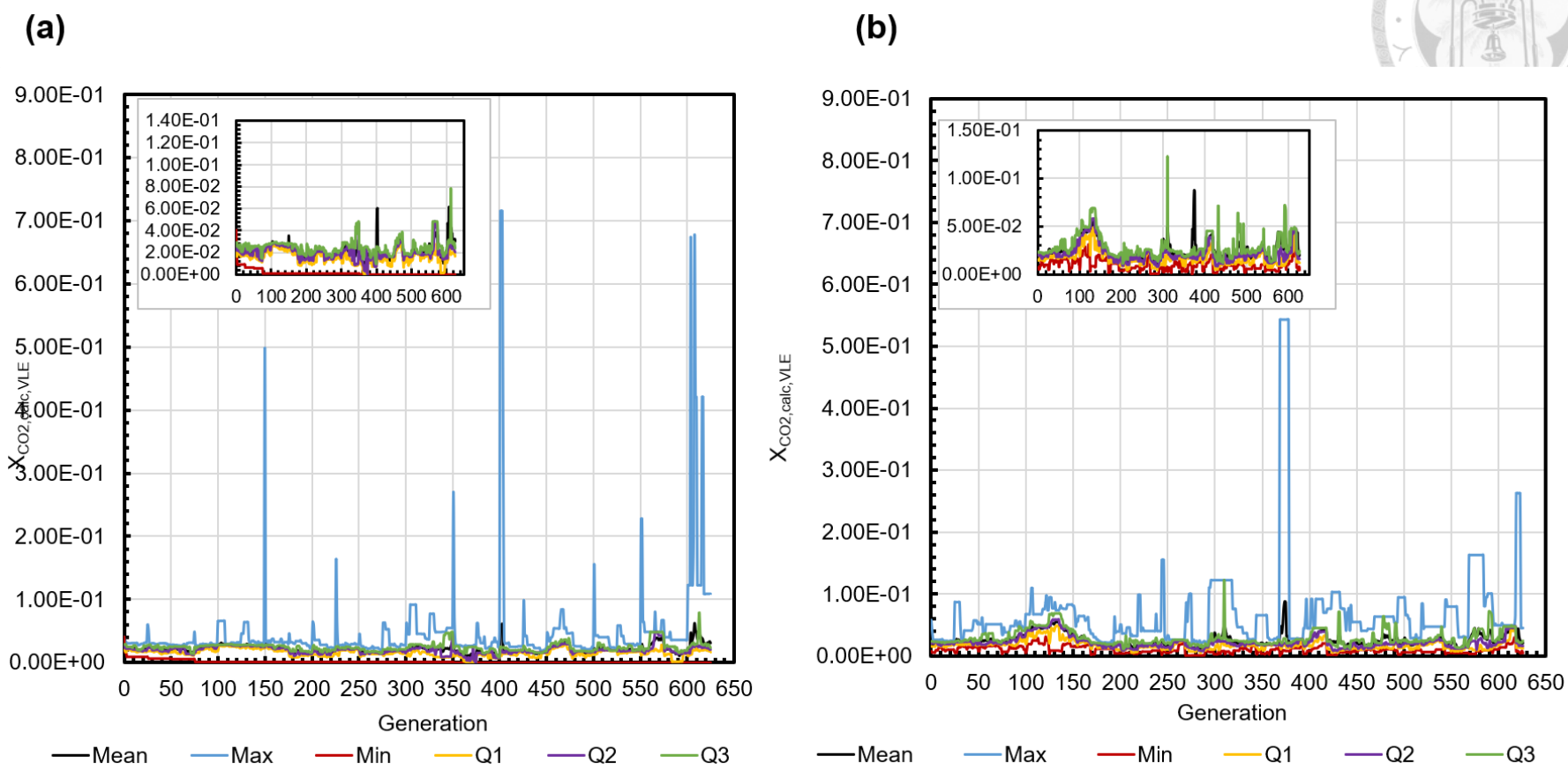


Figure 5.6-1. Evolution trajectory of population in terms of mean, quartiles, maximum, and minimum value of CO_2 solubility for (a) task 1 and (b) task 2.

doi:10.6342/NTU202403528

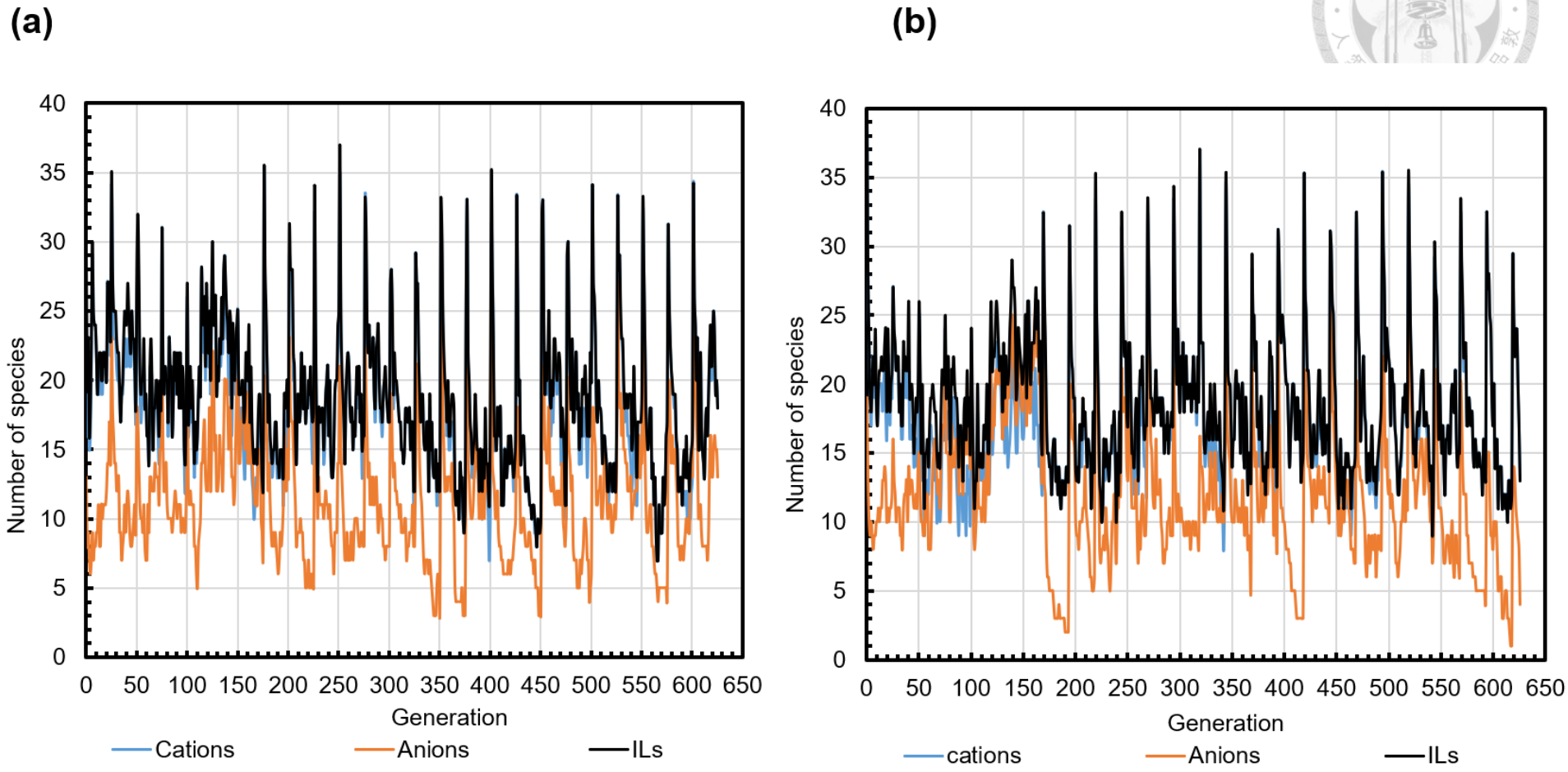


Figure 5.6-2. Evolution trajectory of the number of cation, anion, and IL species existing in population for (a) task 1 and (b) task 2.

doi:10.6342/NTU202403528

The comparison between H_i-derived CO_2 solubility and VLE-based CO_2 solubility for task 1 (3507 IL species) and task 2 (3176 IL species) is shown by **Figure 5.6-3**. Though the CAMD is capable of designing better ILs than specified ILs, most of the new species would only have modest solubility for CO_2. In the regime of $x_{CO2,calc,VLE} < 0.1$, the two methods show good agreement (with $AAD_{Henry\text{-}VLE}$=0.000431 and $AARD_{Henry\text{-}VLE}$=1.43%). However, in the regime of $x_{CO2,calc,VLE} > 0.1$ the $AAD_{Henry\text{-}VLE}$ and $AARD_{Henry\text{-}VLE}$ increase to 0.0631 and 41.2%, respectively. This implies that a minimum $AAD_{Henry\text{-}VLE}$ value of 0.631 might be inevitable for the regime of high CO_2 solubility even though the target of Henry's constant and VLE-based solubility are set 1.00 consistently. Note that the required calculation time for VLE-based method might be significantly longer than that for H_i-based method because of the iterations for the composition in IL phase. The use of Henry's constant might be sufficient if lower accuracy of CO_2 solubility is acceptable.

doi:10.6342/NTU202403528

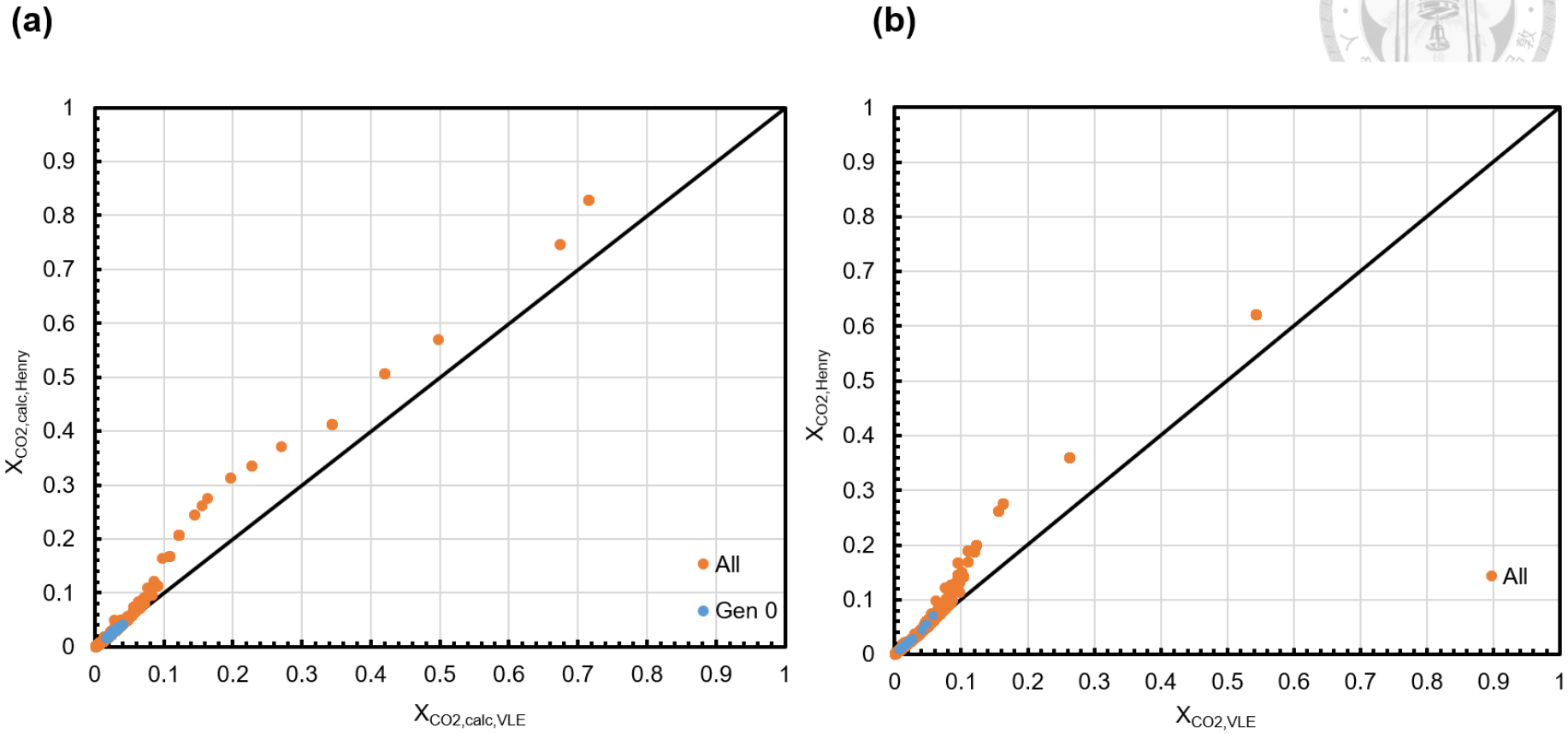


Figure 5.6-3. Comparison between H_i-derived and VLE-based CO_2 solubility in each of the designed IL species in (a) task 1 and (b) task 2. The blue dots are the ILs specified in the initial population, and the orange dots are the designed ILs.

doi:10.6342/NTU202403528

Since the analysis by **Figure 5.6-1** and **Figure 5.6-2** only accounts for the distribution of species in population, the analysis of IL species throughout all generation would be more comprehensive. **Figure 5.6-4** shows the distribution of the accumulated number of IL species with respect to CO_2 solubility per 100 generations, with the inset figure for the regime of higher solution. Note that if an IL species has appeared in former generation, it will no longer be counted in any generation later than its first appearance. Even so, the genetic algorithm is able to produce new ILs species in late generations.

Of all the 3507 IL species generated in task 1, the CO_2 solubility in the 70.34% of the IL species have at least comparable performance with the initial ILs, i.e. higher than the minimum solubility provided by ILs in the initial population, CCCC[n+]1ccccc1.N#C[N-]C#N ($x_{CO_2,calc,VLE} = 0.016487$). However, only 1.11% of all the 3507 IL species are better than the best IL species in the initial population, CNO[N+](CC)(C)C.[F-] ($x_{CO_2,calc,VLE} = 0.058486$). Most of the optimal ILs have halide as its anion part, which has good agreement with experimental findings.

Of all the 3176 IL species generated in task 1, the CO_2 solubility in the 69.54% of the IL species have at least comparable performance with the "lower bound" benchmarking IL, CCCC[n+]1ccccc1.N#C[N-]C#N ($x_{CO_2,calc,VLE} = 0.016487$). However, only 3.46% of all the 3176 IL species are better than the best benchmarking IL, CNO[N+](CC)(C)C.[F-] ($x_{CO_2,calc,VLE} = 0.058486$).

doi:10.6342/NTU202403528

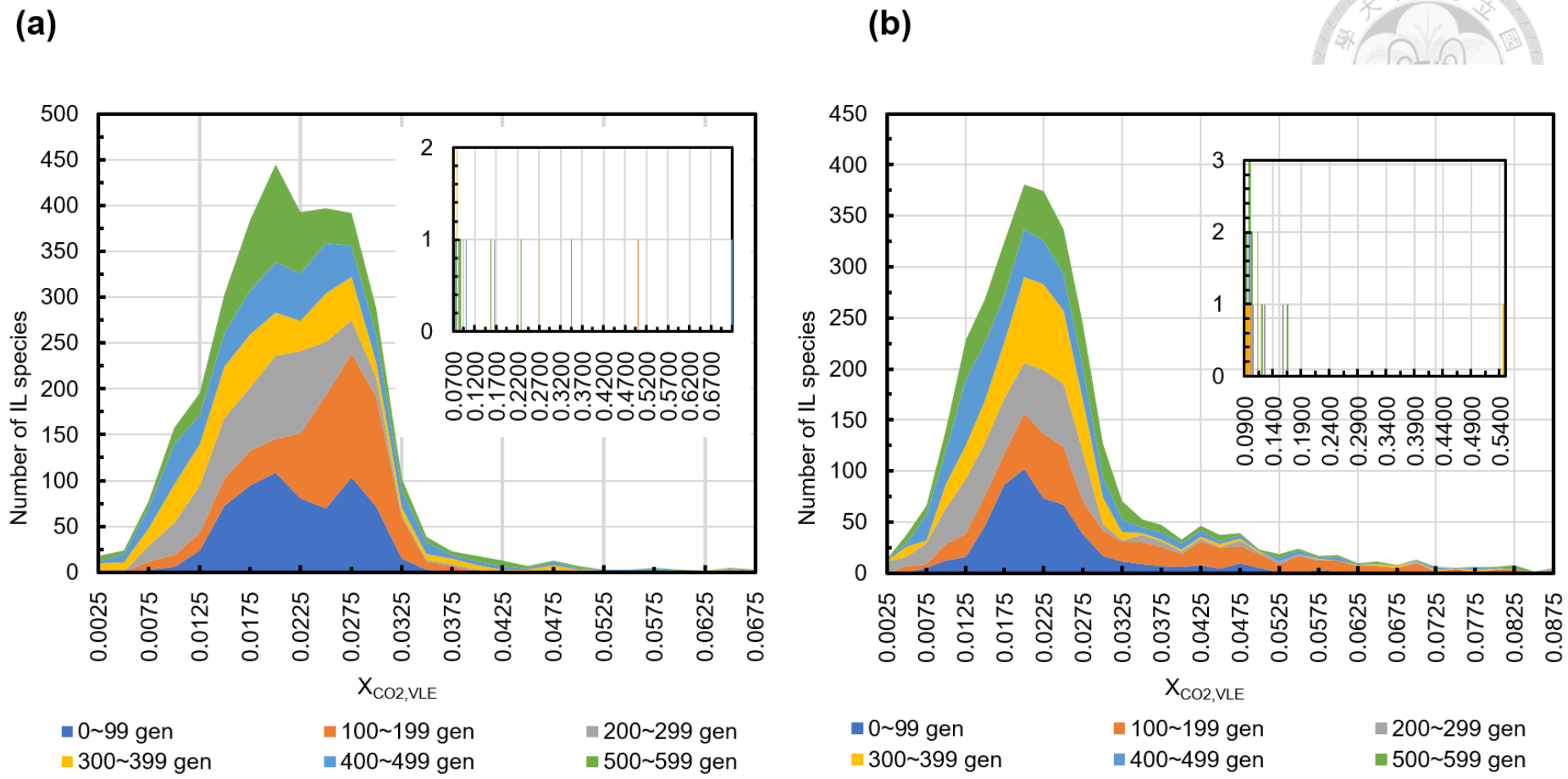


Figure 5.6-4. The distribution of accumulated new ILs species per 100 generations against the CO_2 solubility provided by the ILs in (a) task 1 and (b) task 2. The inset figure shows the regime of higher CO_2 solubility.

doi:10.6342/NTU202403528

Table 5.6-1. The optimal species ($x_{CO2}^{VLE} \geq 0.07$) of task 1

IL	SAscore	SCscore	x_{CO2}^{VLE}
O=N[NH3+].[Cl-]	6.459758	1.730161	7.15130E-01
[NH+]1=C(CC=O)COC1.[OH-]	5.663391	2.257299	6.73963E-01
C[N+](=O)O.[F-]	4.037967	1.308444	4.98330E-01
C[N+]1=C(CO1)CC=O.[OH-]	5.112391	2.146881	4.20333E-01
C[N+](=O)O.[OH-]	4.037967	1.306965	3.43853E-01
N#C[n+]1ccccc1.[F-]	3.868907	1.409937	2.69921E-01
N#C[n+]1ccccc1.[OH-]	3.868907	1.570812	2.27249E-01
[NH+]1=C(CC=O)COCC1.[OH-]	5.344792	2.080685	1.97139E-01
C[n+]1ccccc1.[F-]	2.89633	1.403133	1.63559E-01
C[n+]1ccccc1.[OH-]	2.89633	1.560102	1.55642E-01
O[N+]1=C(CC=O)COCC1.[OH-]	4.657605	2.056511	1.44625E-01
[NH+](=C(CO)CC=O)C.[OH-]	5.808888	1.865491	1.22312E-01
CC(C1=[N+](OC1)C)C=O.[OH-]	5.443628	2.075878	1.08546E-01
O[n+]1ccccc1.[F-]	3.376086	1.202416	9.85660E-02
C[P+]1(C)C#CC1(C)C.[OH-]	5.715484	2.100247	9.13450E-02
ClC(=O)[n+]1ccccc1.[OH-]	3.367023	1.175276	8.56980E-02
OC#C[N+](=O)C#CO.CC[PH-](C(CC)C)(C#CO)(C)C	6.336244	2.574127	8.36600E-02
CC([N+](=C)C)(CC=N)O.[F-]	6.423307	2.309639	8.01830E-02
COCC(=[NH+]C)CC=O.[OH-]	5.333102	2.207268	7.87510E-02
CC(=O)C([NH+](C)C)C.[F-]	5.379002	2.09177	7.68860E-02
O=CC1COCC1=[NH+]C.[OH-]	5.989354	2.216406	7.65240E-02

doi:10.6342/NTU202403528

CC[n+]1ccccc1.[F-]	2.82999	1.423173	7.59730E-02
OC#C[N+](=O)C#CO.CC[PH-](C(C)C)(C#CO)(C)C	6.14283	2.574661	7.25630E-02
CC(=O)C(C1=[N+](OC1)C)C.[OH-]	4.945539	2.334749	7.20170E-02
CO[N+](OO)(OC)C.[OH-]	4.886232	2.375703	7.13900E-02

Table 5.6-2. The optimal species ($x_{CO2}^{VLE} \geq 0.08$) of task 2

IL	SAscore	SCscore	x_{CO2}^{VLE}
O=CC(C=[NH+]C)(C)C.[OH-]	5.840043	1.89082	8.00000E-02
CCC=[N+](OC)C.[F-]	5.185766	2.132104	8.02840E-02
Cc1cc[n+](cc1)C.[F-]	2.886188	1.773975	8.04050E-02
C=CC(=[N+](C(=N)C)C)C.[F-]	6.089354	2.360347	8.07740E-02
Cc1c[n+](C)c2c(c1)C2.[F-]	3.913382	2.152881	8.08520E-02
CCC=C([P+](C(=C=NF)O)(OO)O)C.CCC([PH2-](C)(C)C)(N(C)C)C	6.409714	3.059779	8.10000E-02
Cc1ccc[n+](c1)C.[F-]	3.097125	1.643486	8.12980E-02
FN=C=C([P+](C(=C(N)C)C)(OO)O)O.CCC1(CC[PH-]1(C)(C)CC)N(C)C	6.587133	4.110911	8.13040E-02
FN=C=C([P+](C(=CC)CC)(OO)O)O.CCC([PH-]1(C)(C)CC1)(N(C)C)C(C)C	6.539413	3.548242	8.21360E-02
CCC(=O)C(=[N+](C)C)C.[OH-]	3.909215	1.680957	8.27660E-02
CC(=O)[N+]1=CC#CC1.[OH-]	5.870281	1.819372	8.37440E-02
C[N+](=CC=C)CC#C.[F-]	5.75616	2.064453	8.50460E-02

doi:10.6342/NTU202403528

C=C1CN=N[P+]1(C)C.[F-]	5.836898	2.405142	8.69230E-02
FN=C=C([P+](C(=CF)C)(OO)O)O.CCC([PH-](C)(C)(C)C)(N(C)C)C	6.623562	3.26906	8.69300E-02
FN=C=C([P+](C(=C(C)C)C)(OO)O)O.CCC([PH2-](C)(C)C)(N(C)C)C(C)C	6.235303	3.294852	8.72020E-02
C[N+](=CC#C)C(=O)C.[OH-]	5.411162	1.430787	8.74800E-02
[NH+](N=NC)(CC=O)C.[OH-]	6.721646	1.997546	9.14330E-02
CC(=[N+]1C=CC1=C)C.[F-]	5.058266	2.131142	9.24060E-02
C#C[n+]1ccccc1.[F-]	4.162007	1.429662	9.49730E-02
C#C[n+]1ccccc1.[OH-]	4.162007	1.30316	9.50540E-02
Cc1c2ccc([n+]1C)C2.[F-]	5.403497	2.823614	9.54300E-02
Br[N+](=C)C#C.[F-]	5.880827	1.651061	9.58300E-02
FN=C=C([P+](C(=C)CC)(OO)O)O.CCC([PH-](C)(C)(C)C)(N(C)C)CC	6.294364	3.208405	9.81460E-02
Cc1cccc[n+]1C.[F-]	3.06133	1.597929	9.82020E-02
CC(=O)[n+]1ccccc1.[F-]	3.072457	1.324428	9.90920E-02
C[P+]1(CCN=N1)C=C.[F-]	6.390823	2.106228	1.00680E-01
C[n+]1cc2Cc1cc2.[F-]	5.874027	2.37717	1.01297E-01
C[N+]1=C(C=C)C#CCC1.[F-]	5.712546	2.395702	1.03652E-01
CO[N+]1=NC=CC1=C.[F-]	5.485177	2.148332	1.10217E-01
CO[N+](=O)C#C.[F-]	5.144282	1.5844	1.10712E-01
C=CC(=[N+](C=N)C)C.[F-]	6.508901	2.571599	1.19672E-01
CC#C[N+](=O)OC.[F-]	5.027521	1.69505	1.22925E-01

doi:10.6342/NTU202403528

C[n+]1ccccc1.[OH-]	2.89633	1.560102	1.55649E-01
C[n+]1ccccc1.[F-]	2.89633	1.403133	1.63568E-01
[NH2+]=O.F[P-](F)(F)(F)(F)F	6.449028	1.389601	2.63303E-01
OC=C([PH3+])O.[F-]	5.763995	1.549239	5.43695E-01

doi:10.6342/NTU202403528

Chapter 6. Rule-based vs. AI-based CAMD

6.1. AI-based Generative Models for CAMD

The rapid advancement of artificial intelligence (AI) has witnessed a surge in the development of machine learning (ML) models specifically designed for the field of molecular design.[34, 265-269] Unlike conventional CAMD, AI-based generative models are data-driven and thus require a large database of chemical structures and associated properties. Besides, AI-based molecular design also differs from conventional molecular design in its approaches for generation and representation of chemical structures. The prevailing generative models for Computer-Aided Molecular Design (CAMD) include the RNN-based chemical language model (**Figure 6.1-1** and Appendix F.2)[153, 154, 270-280] and the VAE-based latent variable model (**Figure 6.1-2** and Appendix F.3)[280-292]. Notably, both these models require learning the inherent patterns of valid chemical representations before functioning as generative tools. This stands in contrast to traditional CAMD approaches, where chemical representation is typically predefined before model development. Chemical syntactic models aim to learn the contextual relationships among tokens (e.g. atomic symbols) in sequential data (e.g. SMILES[73, 74]) so that they can generate new sequential data based on the acquired rules. On the other hand, latent variable model aims to create a continuous latent space where a chemical species is represented by a unique numerical vector. This vector encapsulates the abstract chemical patterns of a species. New chemical species can be generated via sampling of points in latent space and can be translated into readable format through decoder. Beyond these prominent model categories, additional machine learning based models have been specifically designed to act as decision-makers for conventional molecular

doi:10.6342/NTU202403528

modifications[291, 293-295], with RDKit employed for the molecular representation and execution of modifications.

Despite these remarkable progresses, both these types of models suffer from some limitations. When these models are trained for exploration tasks, the properties distribution in the generated chemicals, such as logP and SAscores, tends to resemble that of the training data.[284, 296] It is only when these models are trained for exploitation tasks that they exhibit a task-specific distribution. In other words, explorative chemical syntactic models and explorative latent variable models may not be ideal for generating structures with properties beyond the scope of the training dataset. For exploitation tasks, it may be necessary to employ transfer learning[297, 298] to train additional models tailored to different combinations of target properties. Furthermore, these two model types may require additional effort to regularize and rationalize the modification behaviors, such as constraining modifications to practical fragment-scale alterations.[299-301] It may also be challenging for these two models to implement every possible modification at every potential substructure. For example, when atomic symbols are sequentially appended to an existing SMILES, opportunities to connect with inner substructures may be missed. On the other hand, the numerical values in latent vector representation do not directly reveal the actual modification points within a chemical structure, making regulation difficult. In contrast, traditional CAMD approaches and ML-based decision-makers allow for straightforward manipulations of the structure variables in a controllable fashion. Changes to the structure can be realized at the desired resolution, whether it involves the replacement of an atom, a functional group, or an entire structural fragment. The generated molecules are not limited to any prespecified group of molecules.

doi:10.6342/NTU202403528

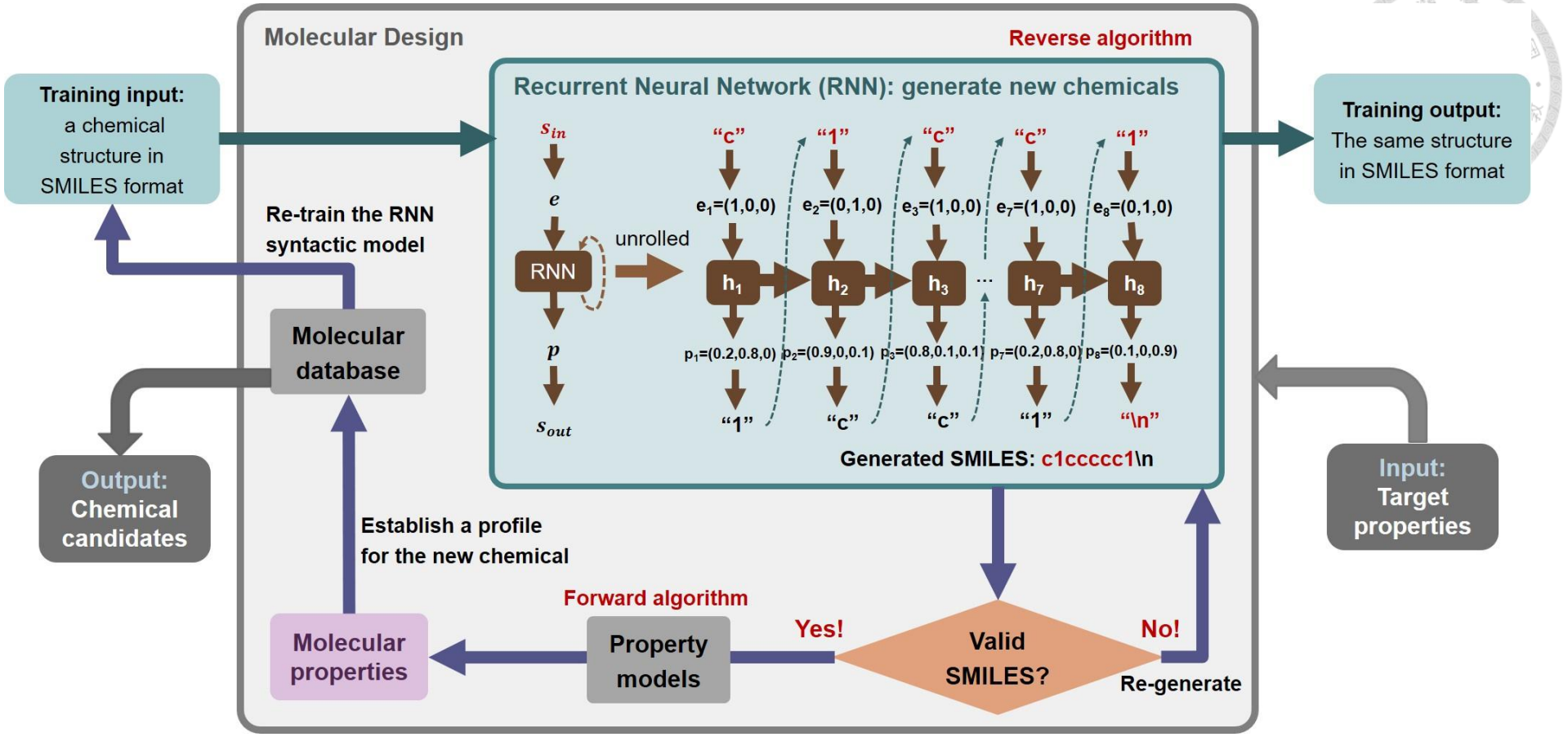


Figure 6.1-1. The flowchart for computer-aided molecular design based on SMILES representation and early architecture of RNN[277].

doi:10.6342/NTU202403528

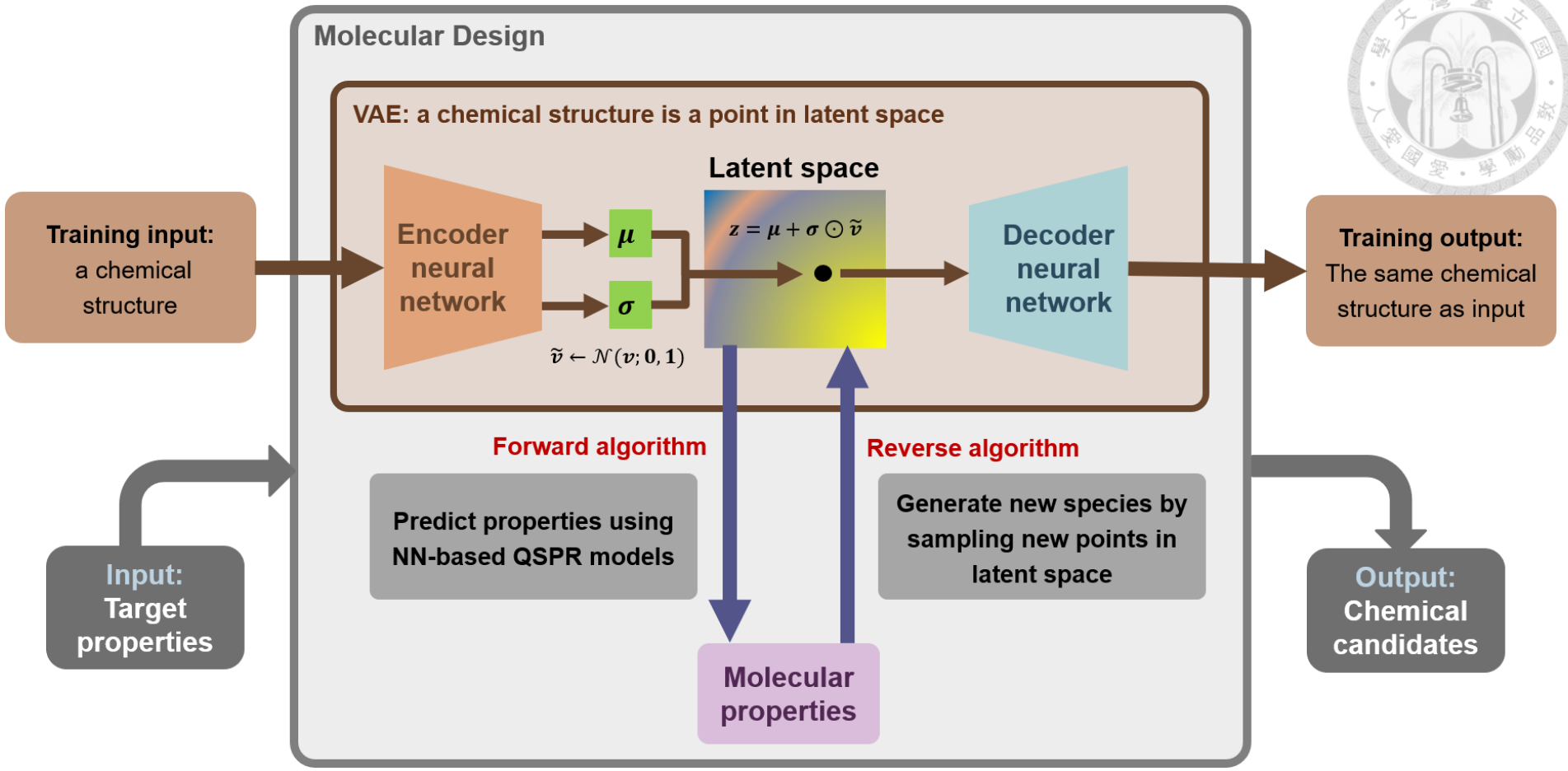


Figure 6.1-2. The flowchart for computer-aided molecular design based on early architecture of VAE[281].

doi:10.6342/NTU202403528

6.2. Benchmarks for Comparing Rule-based and AI-based CAMD

For both chemical syntactic models and latent variable AI models, their performance in exploration tasks[35, 36] are often quantified by **distribution-learning metrics**[111, 302] such as the validity, uniqueness, novelty, and internal diversity of generated chemicals. The significance of these quantities are elaborated in **Table 6.2-1**. In addition, their performance in exploitative design[35, 36] of organic molecules is frequently evaluated by **goal-directed metrics**.[111, 303, 304] These metrics indicate the optimality of the designed molecules in terms of the target values of properties (e.g. logP, SAscore, CNS desirability[305], and the similarity[306, 307] with specific drug-like molecules).

The reconstruction rate[308] of chemicals (see Appendix F.2 and F.3), as indicated by learning loss, is also a crucial metric for assessing the intrinsic accuracy of the model. Both the reconstruction rate and validity from early VAE-based models are often around 50%[275, 283, 287, 289, 309, 310], while for RNN-based models these two metrics are 63.0 to 99.6%[273, 280] and 30.0 to 100.0%[270, 275, 279, 280, 296], respectively. Recent advancements in techniques have led to significant improvements in both types of models. For VAE-based models, the implementation of self-attention transformer[280, 311] (see Appendix F.4) has led to increased chemical validity and reconstruction rates, reaching levels as high as 98.0% to 99.9%[280, 284, 291, 312, 313]. Furthermore, these advancements have also contributed to higher levels of chemical novelty and uniqueness, both exceeding 80.0%[280, 284]. On the other hand, it has been suggested that replacing RNN-based models with transformers[314-316] or introducing attention mechanism to RNN-based models[280] may enhance the chemical validity.

doi:10.6342/NTU202403528

Table 6.2-1. The significance of the distribution-learning metrics.

Metric	Significance
Validity	The fraction of the generated chemicals recognizable by RDKit,[82] indicating how well the model adheres to rules for valid chemical representations.
Uniqueness	The proportion of the distinct chemicals among a set of valid structures, highlighting diversity in generated chemicals.
Novelty	The proportion of the unique structures not represented in the training data, indicating the model's ability to avoid overfitting.
Filters	The proportion of the generated chemicals passing specific filters employed during training data curation (e.g., excluding halogen-containing compounds), demonstrating adherence to additional user-defined constraints.
Internal diversity	The diversity of the generated chemicals measured by Tanimoto similarity, reflecting exploration across different regions of chemical space.

For rule-based generative models, both **goal-directed metrics** are also applicable, whereas certain **distribution-learning metrics** may be less meaningful in some comparisons.[111] Rule-based models are typically developed independently of training datasets and are not designed to replicate the distribution of such datasets. Therefore, the novelty metric, as well as filters metric, will be ill-defined for rule-based generative models. Moreover, robust rule-based models generally ensure the near-perfect validity metric of the generated chemicals for each rule-based model. Validity is often trivial in differentiating the performance of multiple rule-based models.

Currently, numerous benchmark suites are available to facilitate the evaluation of rule-based and AI-based models. Each suite typically includes reference datasets, baseline

doi:10.6342/NTU202403528

generative models (both AI-based and rule-based), distribution-learning metric, and a variety of goal-directed tasks, as detailed in **Table 6.2-2**. These reference datasets serve as training data for baseline models and as initial molecules in goal-directed tasks, aiming to standardize the starting point for each task across different baseline models. This standardization minimizes non-intrinsic differences among baseline models within the same goal-directed task, allowing the intrinsic nature of these generative models to be more clearly characterized by their performance in these tasks. However, achieving a completely fair comparison can be challenging due to different nature of generative algorithms. For example, the approach used by MARS+ to generate new chemicals differ fundamentally from that of RNN-based chemical syntactic models. MARS+ iteratively modifies a population of chemical species to generate new ones, whereas most RNN-based models generate a new species from scratch in one go without modifying existing species.

Two benchmark suites, GuacaMol[111] and MolOpt[317], provide extensive sets of goal-directed tasks and baseline models (detailed in section 6.3 and 6.4), making them popular choices for evaluating the capabilities of new generative models in the exploitative design of organic drug-like molecules. In the following sections, the performance of MARS+-based CAMD in different tasks is ranked with other baseline models based on the two suites.

doi:10.6342/NTU202403528

Table 6.2-2. A survey of benchmark suites and the metrics provided.

Benchmark suite	Benchmark types
GuacaMol[111] ChEMBL 24 dataset	**20 goal-directed tasks**: metrics include similarity with some drug-like molecules, logP, TPSA, and constitutional isomers etc. **Distribution-learning metrics**
MolOpt[317] ZINC 250K dataset	**23 goal-directed tasks**: include most of the goal-directed metrics in GuacaMol plus QED DRD2, GSK3β, JNK3.
Molecular Sets (MOSES)[302] ZINC clean leads dataset	**Distribution-learning metrics**
Tartarus[303] CEPDE & customized datasets	**4 Goal-directed tasks**: metrics include HOMO-LUMO gap, dipole moment, electronic excitation energy, and activation energy.
SMINADockingBenchmark[318] ChEMBL & ZINC datasets	**1 Goal-directed task**: docking affinities serve as the metric
DeNovoBenchmarks[319] GuacaMol & ZINC datasets	**11 Goal-directed tasks**: include a few metrics from GuacaMol and MolOpt such as logP, QED.
MolecularNet[320] QM9, FreeSolv, etc.	Benchmark for property prediction models (e.g., solubility, solvation free energy, lipophilicity) and molecular classification models.

6.3. GuacaMol: Effectiveness of MARS+ and Other Baseline Models

GuacaMol[111] primarily evaluates the effectiveness of generative models in each task. For each goal-directed task, starting chemicals are initialized using the best-suited species from the built-in ChEMBL dataset in GuacaMol. The population size ($|Popu_n|$, see section 3.4.2) is set to 100 and the offspring population size ($|Gen_n|$, see section 3.4.2) is

doi:10.6342/NTU202403528

set to 200. Roulette wheel scheme (section 3.4.2) is adopted for selecting the subject chemical species from population $Popu_n$ for crossover and mutation operations. The probabilities of crossover and mutations are 1.0 and 0.5, respectively. Mutation occurs following successful crossovers. After evaluating the properties of generated species, the most optimal species from the union of the current population and offspring population is selected for the next iteration, i.e. $Popu_{n+1} = \text{find\_optimal}(Popu_n \cup Gen_n)$.

Rule-based models like GRAPH_GA[110, 111] can run up to a maximum of 1000 iterations unless an early-termination criterion is met (i.e. lack of progress in any species within the population over 5 consecutive iterations). Typically, 1000 iterations are sufficient for these tasks to find at least a local optimal species.[317] After the iterations finish, the average score of the top-K species is calculated for particular values of *K*, e.g. the average score of top-1, top-10, and top-100 species. The "score" here shares the same significance with the single-property fitness function used in genetic algorithm based generative models. The overall performance of the generative model in the goal-directed task is quantified by an overall score, determined by the geometric or arithmetic average of these top-K average scores. For instance, using the arithmetic average scheme, the overall score for most goal-directed tasks is represented as:

$$\text{Overall\_score} = \frac{1}{3}(\text{Top1\_avg} + \text{Top10\_avg} + \text{Top100\_avg}) \quad (6.3\text{–}1)$$

In this study, we utilize the 20 **goal-directed tasks** in GuacaMol benchmark suite[111] to compare MARS+ with other baseline models. As mentioned earlier, only metrics related to chemical diversity (i.e., uniqueness and internal diversity) remain well-defined for rule-based models. Since chemical diversity is typically not the primary objective in

doi:10.6342/NTU202403528

practical molecular design tasks, this comparison excludes **distribution-learning metrics**. The target properties for each of the goal-directed tasks are detailed in **Table 6.3-1**, and the baseline models are summarized in **Table 6.3-2**.

Table 6.3-1. Goal-directed tasks in GuacaMol.[111]

Task	Target properties
Celecoxib rediscovery	sim(Celecoxib, ECFC4) = 1
Troglitazone rediscovery	sim(Troglitazone, ECFC4) = 1
Thiothixene rediscovery	sim(Thiothixene, ECFC4) = 1
Aripiprazole similarity	sim(Aripiprazole, ECFC4) $\geq$ 0.75
Albuterol similarity	sim(Albuterol, FCFC4) $\geq$ 0.75
Mestranol similarity	sim(Mestranol, AP) $\geq$ 0.75
C11H24 constitutional isomer	isomer(C11H24) = 1
C9H10N2O2PF2Cl constitutional isomer	isomer(C9H10N2O2PF2Cl) = 1
Median molecules 1	sim(camphor, ECFC4) = 1 sim(menthol, ECFC4) = 1
Median molecules 2	sim(tadalafil, ECFC6) = 1 sim(sildenafil, ECFC6) = 1
Osimertinib MPO	sim(osimertinib, FCFC4) = 1 sim(osimertinib, ECFC6) = 1 TPSA $\geq$ 100 logP $\leq$ 1
Fexofenadine MPO	sim(fexofenadine, AP) $\geq$ 0.8 TPSA $\geq$ 90

doi:10.6342/NTU202403528

	logP $\leq$ 4
Ranolazine MPO	sim(ranolazine, AP) $\geq$ 0.7 logP $\geq$ 7 TPSA $\geq$ 95 number of fluorine atoms = 1
Perindopril MPO	sim(perindopril, ECFC4) = 1 number aromatic rings
Amlodipine MPO	sim(amlodipine, ECFC4) number rings = 2
Sitagliptin MPO	sim(sitagliptin, ECFC4) = 0 logP = 2.0165 TPSA = 77.04 isomer(C16H15F6N5O) = 1
Zaleplon MPO	sim(zaleplon, ECFC4) = 1 isomer(C19H17N3O2) = 1
Valsartan SMARTS	SMARTS(s_2) = 1 logP = 2.0165 TPSA = 77.04 Bertz = 896.38
deco hop	SMARTS(s_2) = 1 SMARTS(s_3) = 0 SMARTS(s_4) = 0 sim(s_5, PHCO) $\geq$ 0.85
scaffold hop	SMARTS(s_2) = 0 SMARTS(s_6) = 1

doi:10.6342/NTU202403528

	sim(s_5, PHCO) $\geq$ 0.75

†s_2 to s_6 are chemical patterns in reaction SMARTS format.

s_2: [#7]-c1n[c;h1]nc2[c;h1]c(-[#8])[c;h0][c;h1]-c12

s_3: [#7]-c1ccc2ncsc2c1

s_4: CS([#6])(=O)=O

s_5: CCCOc1cc2ncnc(Nc3ccc4ncsc4c3)c2cc1S(=O)(=O)C-(C)(C)C

s_6: [#6]-[#6]-[#6]-[#8]-[#6]~[#6]~[#6]~[#6]~[#6]-[#7]-c1ccc2ncsc2c1

Table 6.3-2. The five baseline models in GuacaMol.[111]

Baseline model	Generative algorithm
BEST_FROM_DATASET	The best species screened from the built-in dataset.
GRAPH_GA	Genetic algorithm-based optimization molecular graphs. It is similar to MARS+.
SMILES_LSTM_HC	RNN-based chemical syntactic model based on SMILES (Appendix F.2)
SMILES_GA	Genetic algorithm-based optimization on SMILES.
GRAPH_MCTS	RNN-based chemical syntactic model (Appendix F.2) based on SMILES, with Monte Carlo tree search algorithm for species generation.

The comparison between our MARS+ based CAMD and other baseline models are presented in **Figure 6.3-1**. Except for the search for constitutional isomers of C11H14 and C9H10N2O2PF2Cl, MARS+ ranks the second, following GRAPH_GA. In most tasks, MARS+ shows slightly compromised performance compared to GRAPH_GA, although its overall performance (i.e., the sum of overall scores across tasks) is comparable to GRAPH_GA. The difference between MARS+ and GRAPH_GA is particularly notable in the task of searching for constitutional isomers of C11H14 and

doi:10.6342/NTU202403528

C9H10N2O2PF2Cl. We suspect this discrepancy may stem from inefficient fragment exchange in our crossover operation (see **Figure 3.3-2**). From the source code of GRAPH_GA, it appears that all base elements in a fragment can connect with another fragment in GRAPH_GA's crossover operation if their valences are compatible, whereas in our crossover operation, a fragment can only connect with another one at the crossover point. We attempted to align our crossover operation with GRAPH_GA's implementation, resulting in a revised version called *MARS+_modcross*. The performance of *MARS+_modcross* is also shown in **Figure 6.3-1**. This revision significantly improves MARS+'s performance in the tasks involving C11H14 and C9H10N2O2PF2Cl. However, it also leads to substantial performance sacrifices in some single-objective tasks such as Celecoxib rediscovery and Troglitazone rediscovery, and a slightly compromised performance in other tasks.

doi:10.6342/NTU202403528

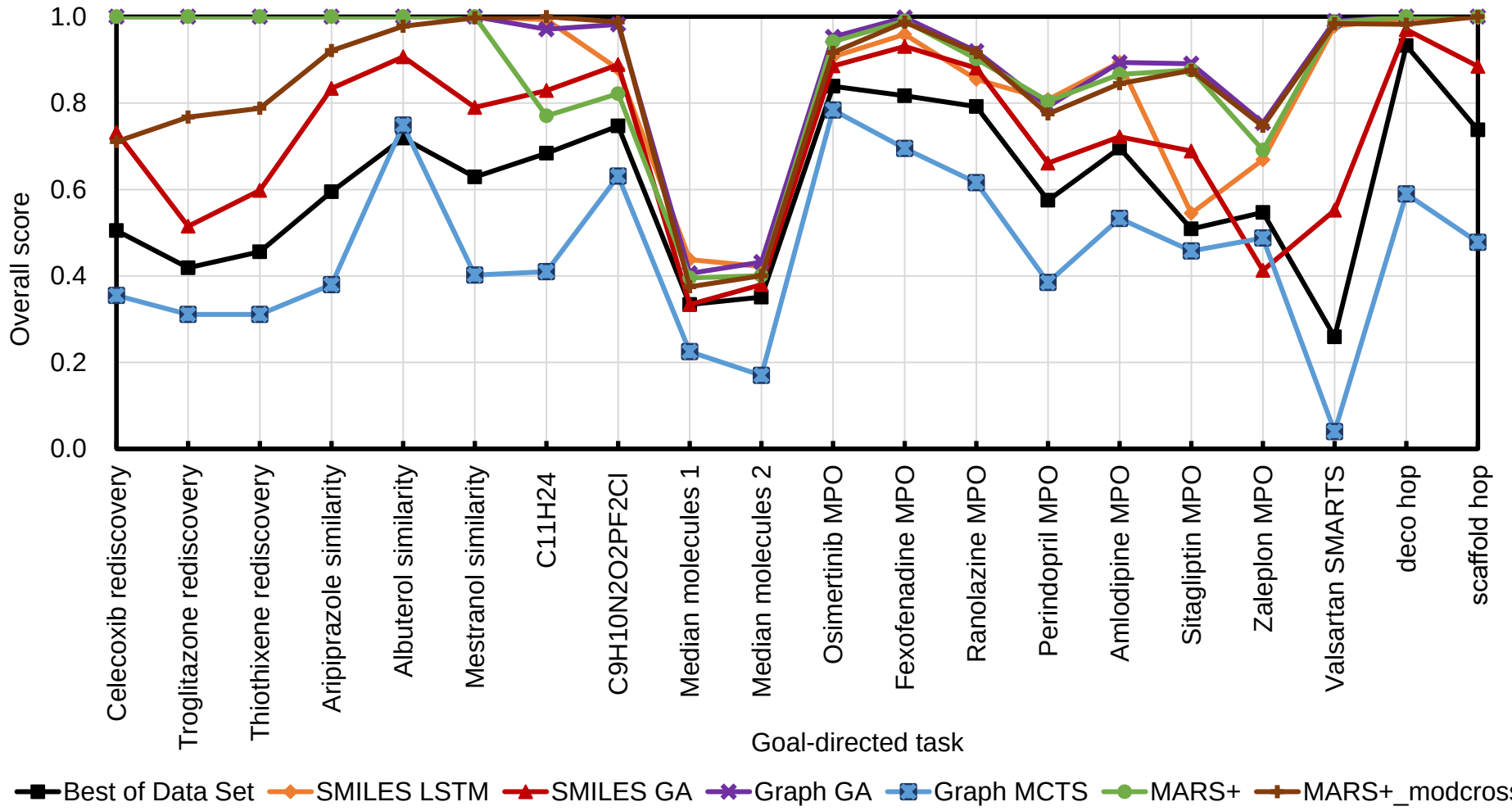


Figure 6.3-1. Performance of 5 baseline models in GuacaMol benchmark, MARS+, and MARS+_modcross.

doi:10.6342/NTU202403528

6.4. MolOpt: Efficiency of MARS+ and Other Baseline Models

In contrast to GuacaMol, MolOpt[317] assesses the efficiency of generative models in achieving a goal-directed task. Each task begins with randomly selected chemicals from the built-in ZINC 250K dataset. The population size ($|Popu_n|$ section 3.4.2) is set 120 and the offspring population size ($|Gen_n|$ section 3.4.2) is set 70. Roulette wheel scheme (section 3.4.2) is employed to choose chemical species from population $Popu_n$ for crossover and mutation operations, with crossover and mutation probabilities set at 1.0 and 0.067, respectively. Mutation occurs subsequent to successful crossovers. Following evaluation of the generated species' properties, the most optimal species from the union of the current population and offspring population is selected for the subsequent iteration, i.e. $Popu_{n+1} = \text{find\_optimal}(Popu_n \cup Gen_n)$.

All hyperparameters of AI models and parameters of rule-based models are fine-tuned to optimize the AUC-top10 metric (detailed in next paragraphs) in Zaleplon MPO and Perindopril MPO tasks (see **Table 6.3-1**). Rule-based models such as GRAPH_GA[110, 111] are allowed to evaluate up to a maximum of 10,000 new unique species unless an early-termination criterion is met (i.e. lack of progress in the average score of the population over 5 consecutive iterations). Upon generating *i*-th new unique species and evaluating its properties, the average score of the top-K species at this stage is computed, as shown in eq (6.4–1).

$$\text{TopK\_avg}(\boldsymbol{u_1}, \boldsymbol{u_2}, \dots, \boldsymbol{u_i}) = \frac{1}{K}\sum_{j=1}^{K} \text{scoring\_func}(\boldsymbol{u_{j/i}}) \qquad (6.4\text{–}1)$$

Here, $\boldsymbol{u_i}$ represents the *i*-th generated species, and $\boldsymbol{u_{j/i}}$ is the *j*-th best species in

doi:10.6342/NTU202403528

the generated species $\{\boldsymbol{u_1}, \boldsymbol{u_2}, \ldots, \boldsymbol{u_i}\}$. If $K > i$, then only the i available species, $\{\boldsymbol{u_1}, \boldsymbol{u_2}, \ldots, \boldsymbol{u_i}\}$, are used in the calculation top-K average score. The overall efficiency of the generative model in the goal-directed task is quantified using the area under top-K curve (AUC-topK)[317, 321], described by eq (6.4–2). Given that each top-K average score ranges between 0 and 1, the constant 1/10000 normalizes the AUC-topK to the closed interval $[0, 1]$.

$$\text{AUC\_topK} = \frac{1}{10000}\int_1^N \text{TopK\_average}(\boldsymbol{u_1}, \boldsymbol{u_2}, \ldots, \boldsymbol{u_i})di \tag{6.4–2}$$

This study compares MARS+ with other baseline models across 18 goal-directed tasks from the MolOpt benchmark suite[317], namely, all the tasks in **Table 6.3-1** with exclusion of C11H24 isomer searching and Aripiprazole rediscovery. Details of the baseline models in MolOpt are summarized in **Table 6.4-1**.

Table 6.4-1. The 25 baseline models in MolOpt.[317]

Baseline model	Generative algorithm
GRAPH_GA	Rule-based and optimization-based genetic algorithm on molecular graphs. It is similar to MARS+.
REINVENT	RNN-based chemical syntactic model based on SMILES (Appendix F.2)
REINVENT_SELFIES	RNN-based chemical syntactic model based on SELFIES (Appendix F.2)
GP_BO	GRAPH_GA with GP_BO optimization algorithm
STONED	Genetic algorithm-based optimization on SELFIES.

doi:10.6342/NTU202403528

SMILES_LSTM_HC	RNN-based chemical syntactic model based on SMILES (Appendix F.2)
SMILES_GA	Genetic algorithm-based optimization on SMILES.
SYNNET	Reaction-based optimization on molecular graph with neural network model for decision making.
DOG_GEN	Reaction-based optimization on molecular graph with RNN model for decision making.
DST	Genetic algorithm-based optimization on GNN-based molecular graph
MARS	Genetic algorithm-based optimization on GNN/MPNN-based molecular graph
MIMOSA	Genetic algorithm-based optimization on GNN-based molecular graph
MOL_PAL	Property model-based (MPNN) screening method
SELFIES_LSTM_HC	RNN-based chemical syntactic model based on SELFIES (Appendix F.2)
DOG_AE	Reaction-based optimization on molecular graph with RNN model for decision making and autoencoder as an extra molecular representation for new species generation.
GFLOWNET	Genetic algorithm-based optimization on GNN-based molecular graph
SELFIES_GA	Genetic algorithm-based optimization on SELFIES.
SELFIES_VAE_BO	VAE-based latent variable model based on SMILES (Appendix F.3)
SCREENING	Randomly sampling from ZINC 250K reference dataset.
SMILES_VAE_BO	VAE-based latent variable model based on SMILES (Appendix F.3)
PASITHEA	Direct gradient-based molecule optimization employing

doi:10.6342/NTU202403528

	DNN and SELFIES
GFLOWNET-AL	GFLOWNET with extra property models included to enhance sampling efficiency
JT_VAE_BO	VAE-based latent variable model based on SMILES (Appendix F.3)
GRAPH_MCTS	RNN-based chemical syntactic model (Appendix F.2) with Monte Carlo tree search
MOLDQN	Reaction-based optimization on molecular graph with neural network model for decision making.

The comparison of our MARS+ with other baseline models is depicted in **Figure 6.4-2**. In terms of overall performance, measured by the sum of AUC top-K scores across tasks, *MARS+_modcross* ranks 3rd, following REINVENT (1st) and GRAPH_GA (2nd). Similar to the findings in GuacaMol, *MARS+_modcross* demonstrates performance comparable to GRAPH_GA in most of the tasks, except for Celecoxib rediscovery and the search for C9H10N2O2PF2Cl isomers. In particularly, *MARS+_modcross* exhibits apparent inefficiency in Celecoxib rediscovery. Upon examination of the generated species, it appears that high-scoring species from GRAPH_GA, such as Cc1ccc(c2cc(N)c(=O)n(C(C)C(=O)NC3CCCCC3)n2)o1 with a score of 0.868, typically contain abundant cyclic substructures.

Our crossover operation appears to favor the disruption of cyclic substructures, as mentioned in section 3.3.2 that rings will be destructed if unpaired cyclic flags arise during the fragmentation process. In contrast, GRAPH_GA has a ring crossover mechanism (detailed in **Figure 6.4-2**), which exchanges ring components between two molecules while ensuring that the number of rings in each molecule remains unchanged after operation. This mechanism could be seen as an extension of our crossover approach,

doi:10.6342/NTU202403528

where the second cut is always on a ring bond in our current implementation. It would be beneficial to generalize our crossover mechanism in future work to incorporate similar ring-preserving strategies seen in GRAPH_GA, potentially enhancing the efficiency of cyclic structure preservation in molecule design tasks.

REINVENT emerges as the most effective model across a majority of tasks, highlighting the superior efficiency of its reinforcement learning-based training algorithm compared to other models utilizing similar working principle, such as SMILES_LSTM_HC.[317] In contrast, methods that rely only on the incremental construction of molecules from a single starting point using small building blocks (e.g. tokens or atoms), such as MOLDQN and GRAPH_MCTS, prove to be less efficient. While these approaches have the potential to explore a broader chemical space, they are more suited for explorative tasks rather than exploitative ones.

doi:10.6342/NTU202403528

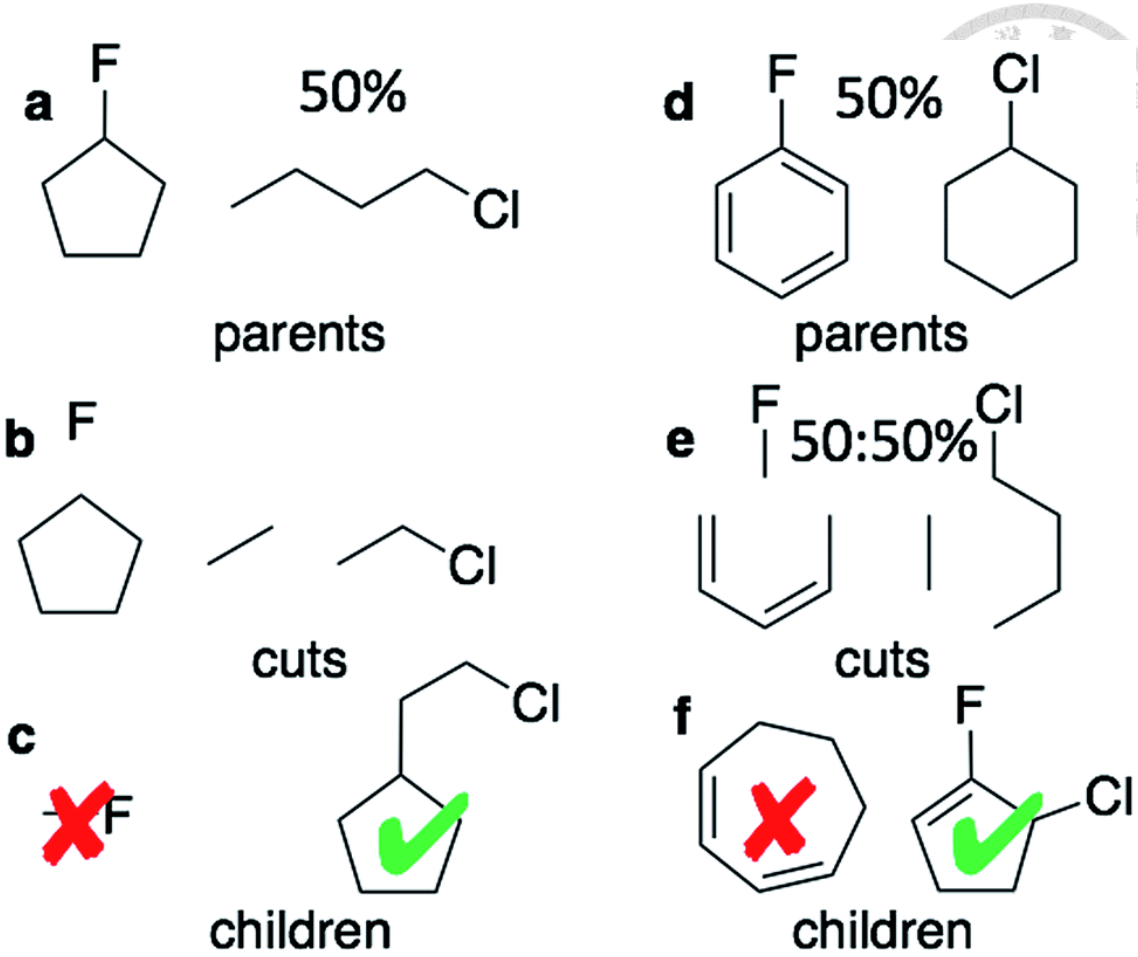


Figure 6.4-1. The steps of (a-c) plain crossover and (d-f) ring crossover in GRAPH_GA. Ring crossover requires two cuts: one at a specified bond and another at adjacent bonds or bonds separated by one bond. In step (f), the resulting ring fragments from step (e) are paired and connected using ring bonds to ensure the number of rings in each molecule remains unchanged after operation. This figure is reproduced from reference[110] with permission from the Royal Society of Chemistry.

doi:10.6342/NTU202403528

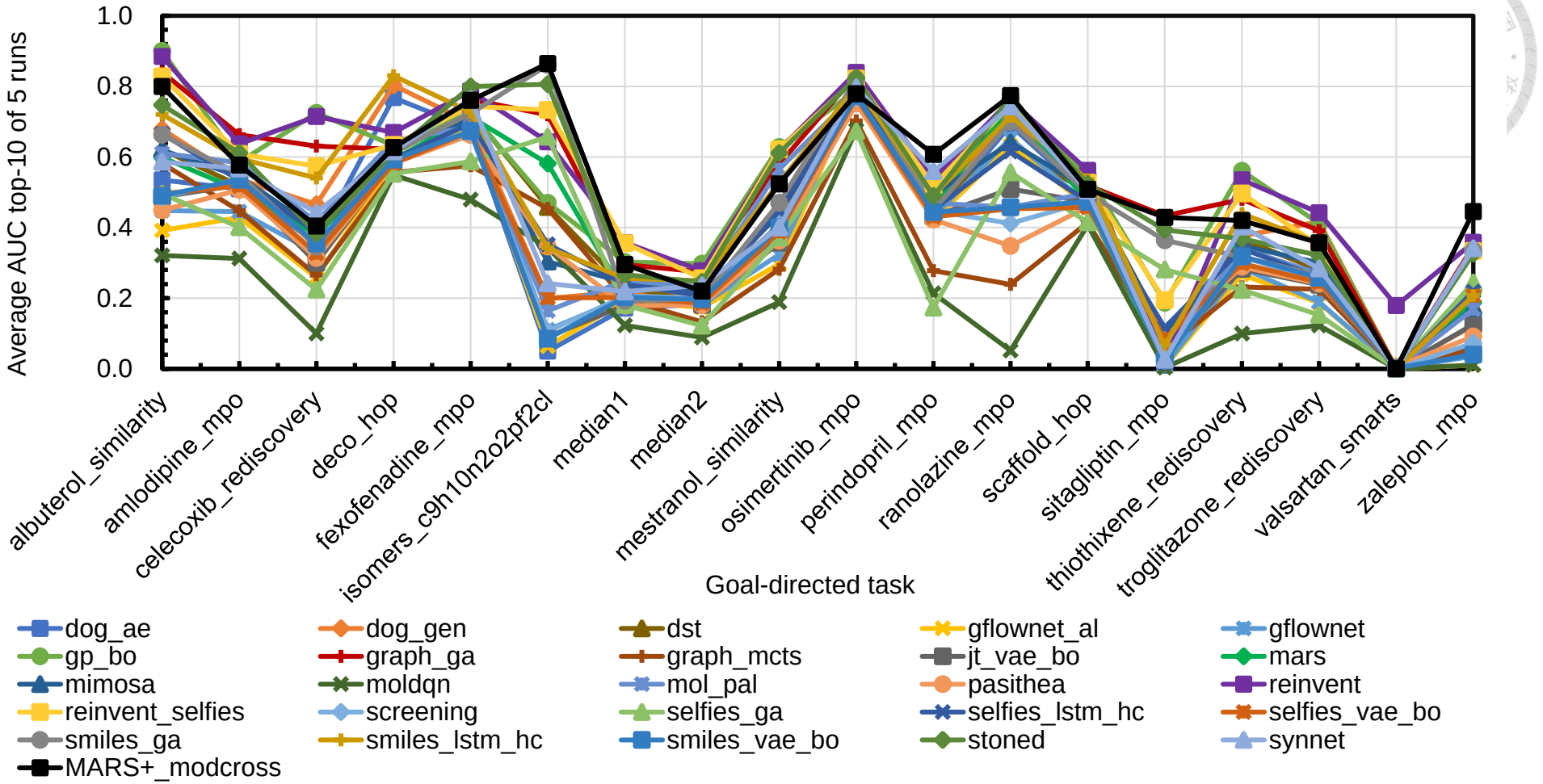


Figure 6.4-2. Performance of 25 baseline models in MolOpt benchmark and MARS+_modcross.

doi:10.6342/NTU202403528

Chapter 7. Conclusions

The previously developed molecular design tool, MARS[162], is extended to handle various types of chemicals, including linear, branched, cyclic compounds, ionic species, cis-trans isomers, and optical isomers. The new program, MARS+, has the following outstanding features:

(1) 12 reversible operations for molecular structure manipulation

(2) Each operation can be applied on all possible positions (atoms or bonds)

(3) Capable of handling isomers (enantiomers and cis-trans isomers)

(4) Capable of handling neutral and ionic species

(5) Most molecular structures can be generated with a series of available operations

As a traditional method, MARS+ can generate new chemical species without an existing structural database, and the structure of generated chemicals can be very different from any existing ones. traditional method also offer advantages of CAMD in that the structure change can be applied to a desired position and in a desired fashion. This is different from most RNN-based chemical syntactic models and VAE-based latent variable models, where the learned molecular representation may not be sufficiently robust for that. In addition, genetic algorithm-based CAMD has been demonstrated to be comparable to, or even better than, ML-based models in certain well-studied tasks, such as drug design. We illustrate that MARS+ is capable of generating very complex structures by a combination of molecular operations. A very large database of chemicals can be easily created by extensive and exhaustive repetitions of all operations on every site of any existing molecule. Such a database with rich structure diversity and high

doi:10.6342/NTU202403528

resolution may also be useful for developing data-driven methods.

MARS+ is applied to the design of novel ILs as CO_2 absorbents. The potential advantages of ionic liquids over the conventional solvents in CO_2 capture have drawn a lot of attention. For the search of specialty ILs, the combinatorial screening of common cations and anions is often used. Typically, such method can generate the ILs more feasible for laboratory synthesis, though the pre-defined library of ions largely limits the variety of IL candidates. In this work, we show that the atomic level CAMD can compensate for the disadvantages of screening methods. The proposed CAMD framework is able to automatically create numerous ions based on genetic algorithm, though the feasibility of laboratory synthesis for ILs is not as satisfactory as that from screening method. Under the given target values and parameter settings, the results show that the specification of initial population has limited influence on the design of ILs with high CO_2 solubility ($x_{CO_2} > 0.1$). However, it will help the design of ILs with better similarity and comparable CO_2 solubility.

Finally, MARS+ is benchmarked against other baseline models using the GuacaMol and MolOpt suites. In the effectiveness test with GuacaMol, MARS+ ranks second among 5 models, just behind GRAPH_GA. Across most goal-directed tasks, MARS+ performs comparably to GRAPH_GA, except in the search for constitutional isomers of C11H24 and C9H10N2O2PF2Cl. Generalizing the crossover operation could potentially improve performance in these tasks, but it may come at the cost of performance on other single-objective tasks, such as Celecoxib and Troglitazone rediscovery. In the efficiency test with MolOpt, MARS+ ranks third out of 26 models, following REINVENT (1st) and GRAPH_GA (2nd). Notably, MARS+ exhibits particular inefficiency in Celecoxib rediscovery compared to GRAPH_GA, which may be attributed to the absence of a ring crossover mechanism in MARS+.

doi:10.6342/NTU202403528

Chapter 8. Prospects and Future Work

8.1. Applications to Other Chemical Mixture Systems

The current version of MARS+ demonstrates potential for handling various complex systems with minimal or moderate modifications to its source code. These systems encompass pharmaceutical cocrystals[322], double-salt ionic liquids (DSILs)[58], deep eutectic solvents (DESs)[323, 324], optoelectronic materials[2, 325, 326], biomolecules[327], and polymers[328-331].

Pharmaceutical cocrystals are crystalline materials typically composed of an active pharmaceutical ingredient (API), either neutral or ionic, and a neutral coformer. The coformer significantly influences key physicochemical properties of the cocrystal, such as melting point, chemical stability, solubility, dissolution rate, and bioavailability. This influence stems largely from hydrogen-bonding interactions between the API and the coformer. Moreover, the coformer can potentially alter the structural integrity of the API, thereby modifying its properties. Designing optimal conformers for drug delivery and release would be a valuable area of research.

Double-salt ionic liquids (DSILs) and deep eutectic solvents (DESs) are natural extensions of the current study. DSILs consist of two distinct ionic liquids (ILs), [Cat1][An1] and [Cat2][An2], blended in a specific stoichiometric ratio. This mixture system introduces additional combinatorial degrees of freedom in its components, making it an ideal subject for demonstrating the capabilities of CAMD and screening methods. DESs are often regarded as a specialized category within ILs. Their unique characteristic lies in the hydrogen-bonding interactions among their components, which leads to unexpectedly deep depression in their melting points.

doi:10.6342/NTU202403528

Our implementation of MARS+ based CAMD incorporates key performance properties of optoelectronic materials, including fundamental gap, electron affinity, ionization potential, and electronegativity (see section 3.2). Future enhancements may involve integrating properties such as exciton binding energy[2], which could serve as a preliminary indicator for assessing a molecule's suitability for applications in photovoltaic cells or light-emitting devices.

The design of biomolecules and polymers is theoretically achievable using MARS+. However, efficient conformer enumeration and geometry optimization are often necessary for these studies, as the physicochemical properties of macromolecules can be highly dependent on their conformations. Performing force field-based atomic optimization using RDKit and OpenBabel can be time-consuming for each macromolecule. Fortunately, AlphaFold[332, 333] can efficiently handle this task, significantly reducing the computational time required.

8.2. Enriching the Mechanisms for Molecular Manipulations

Up to this point, we should highlight the distinctions between MARS+ and the other traditional atom-based and fragment-based algorithms. MoleculeEvoluator[109] (proprietary), Spaceship[108] (proprietary), Molpher[106, 107] (open-source), EvoMol[112] (open-source), and GraphGA[110, 111] (open-source) are software for atom-based molecular design. While MoleculeEvoluator[109] and GraphGA[110, 111] bear some similarities with MARS+ in terms of the overall scheme of crossover and mutation operators, they do not appear to explicitly consider the operators for merging and isomerism inversions. MARS+ also provides more flexibility for operators. For instance, MoleculeEvoluator[109] only uses single bond for “ring creation”, while MARS+ permits the use of single, double, and triple bonds for that. The “uninsert atom” in MoleculeEvoluator[109] is applicable only for the

doi:10.6342/NTU202403528

atoms exactly connecting with two heavy atoms, whereas MARS+ allows the *subtraction* of any atom as long as the junction valences of the two remaining subgraphs are compatible. On the other hand, as the atom-based modifications of GraphGA[110, 111] are implemented via reaction SMARTS, additional (yet trivial) reaction templates may be required to extend the operator's applicability to more substructures. For example, in an exploration task using GraphGA[110, 111], one may need a template for each increment in ring size to facilitate the formation of 7-membered and larger rings. However, the formation of these large rings is naturally feasible in MARS+.

Molpher[106, 107] and EvoMol[112] also employ atom-based modifications, but they lack mechanisms for ring formation/destruction, crossover, merging, and isomerism inversions. Spaceship[108] shares similarities with these two, yet it can introduce aromatic rings through a mutation mechanism. These three software propose a "bond rearrangement" (or "group moving") mechanism that can relocate a side chain within a chemical structure. The fragment-based LEADD[113] (open-source) also propose interesting modification mechanisms such as "internal expansion" and "translation". In "internal expansion" operation, a subject atom/fragment in a chemical structure is self-cloned to form a replica as its additional connecting neighbors, and then the subject atom/fragment is replaced with another type of atom/fragment. Although MARS+ lacks an "internal expansion" mechanism, it can still achieve similar modifications through consecutive operations of *addition* and *change_element*. On the other hand, the "translation" operation is analogous to the "bond rearrangement" mentioned earlier but emphasizes the explicit potential to rearrange atoms/fragments to the inner points of a chemical graph via insertions and uninsertions. Given that "translation" could be especially useful in exploring constitutional isomers, it is worth noting that MARS+ should consider its inclusion in future work.

doi:10.6342/NTU202403528

The majority of fragment-based algorithms have simpler connection rules and modification mechanisms compared to atom-based algorithms. For instance, the available valences of a fragment in LigBuilder[116, 117] are restricted those originally bonded with a hydrogen atom, and therefore the connection between two adjacent fragments is always a single bond. CReM[105] (open-source) provides addition of fragments to a molecule, substitution of a substructure with a fragment, and merging of molecules with linkers. They do not propose mechanism such as *change_element* or *subtraction,* as these are generally less necessary for their specific purposes. On the other hand, Flux[60, 61] demonstrates the use of retrosynthesis techniques to enhance the synthetic feasibility of generated molecules.

8.3. Integrated Computational Molecular-Process Design

Since the designed chemicals ultimately need to be applied to practical processes to evaluate their impacts, it is advantageous to incorporate process objectives and constraints into the molecular design task, forming the integrated computational molecular-process design[334]. Process objectives and constraints can be directly integrated into the equality constraints $\boldsymbol{h}(\boldsymbol{u_i}, \boldsymbol{w_i}) = 0$ and inequality constraints $\boldsymbol{g}(\boldsymbol{u_i}, \boldsymbol{w_i}) \leq 0$ in the mixed-integer nonlinear programming (MINLP) formulation, with process variables $\boldsymbol{w_i}$ optimized simultaneously with chemical species $\boldsymbol{u_i}$.[335-337] Alternatively, the integrated molecular-process design can be divided into two stages: the first stage focuses on computational molecular design, while the subsequent stage optimizes the process with chemical species fixed to those identified as optimal in the first stage.[338, 339] This alternative approach is similar to the GBD method (see Appendix E) in solving MINLP problem.

doi:10.6342/NTU202403528

8.4. Qualitative Comparisons for Implemented Selection Algorithms

In section 3.4, we have implemented several selection algorithms within MARS+ based CAMD. Conducting a qualitative comparison to analyze the behaviors of these algorithms would provide valuable insights. Such comparisons could serve as a basis for selecting appropriate algorithms for more complex goal-directed tasks encountered in future application studies, and also guide improvements to these algorithms. We designed two goal-directed tasks to characterize these selection algorithms: one simple and one challenging.

In both tasks, logP is selected as the sole target property, with the initial population consisting of molecules having logP values between 8.0 and 9.0. The simple task sets the target logP to 4.0, while the challenging task sets it to -4.0. AUC top-K (see section 6.4) may be utilized to measure the efficiency of each selection algorithm.

It is important to note that this comparison differs from the studies in sections 6.3 and 6.4. Here, the comparison is conducted with a fixed choice of molecular data structure (MDS) and a fixed generative algorithm, specifically MARS+. Sections 6.3 and 6.4, in contrast, compare interplaying effects across different combinations of MDSs, generative algorithms, and selection algorithms.

doi:10.6342/NTU202403528

Appendix A. Supplementary Tables to the Main Texts

Table A1. A survey of molecular and property databases. Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.

Database/Research	Covered chemical subspace	Properties
GDB-17[15-18]	~ 1.7×10^{11}	HOMO, LUMO, dipole, IP, EA etc.
NIST CCCBDB database[340]	~ 2.1×10^{3}	
ChemSpider[13]	~ 7.1×10^{7}	MW, T_m, T_b, P^{vap}, toxicity etc.
PubChem database[14]	~ 2.5×10^{8}	
NIST chemistry Webbook[341]	~ 7.3×10^{4}	Thermophysical data, MW, T_m, T_b, toxicity etc.
Dortmund databank[19, 20]	Pure: ~ 8.7×10^{4} species VLE: ~ 4.4×10^{4} mixtures LLE: ~ 4.1×10^{4} mixtures etc.	Thermophysical data (e.g. P^{vap} ,VLE, LLE, etc.), MW, T_m, T_b
Beilstein database[12]	~ 1.0×10^{7}	Reaction mechanisms
LOLI database (ChemADVISOR)[11]	More than 6.0×10^{5} species	Regulatory data (e.g. medical, toxicological, pharmacological and clinical data etc.)

doi:10.6342/NTU202403528

Table A2. A survey of chemical space exploration and size estimation research.

Reference	Conditions	Estimated size of chemical subspace
Drew, K. L. et al.[26]	(1) Number of C atoms ≤ 100 (2) Consists of C, N, O, F, P, S, Cl, Br, I and H	Organic molecules: 3.4×10^{9} Drug-like molecules: 1.5×10^{7}
Weininger, D.[29]	(1) MW ≤ 1000 g/mol (2) Consists of C, N, O, F, P, S, Cl, Br, I and H (3) Consider stereoisomers	10^{180}
Bohacek, R. S. et al. [29, 342]	(1) MW ≤ 500 g/mol, heavy atoms ≤ 30 (2) Consists of C, N, O, F, S, Cl, Br, and H	10^{63}
Walters, W. P. et al.[29, 343]	Virtual screening based on the existing building blocks in typical combinatorial libraries	10^{100}
Lemonick, S.[344]	The number of organic and inorganic substances in the CAS database	10^{8}
Ogata et al.[27]	Consists of C, N, O, S, Cl and H	$10^{8} \sim 10^{19}$

doi:10.6342/NTU202403528

Table A3. A survey of work applying CAMD for chemical engineering problems*. Reprinted with permission from the reference[264]. Copyright 2018 American Chemical Society.

Property estimation / MINLP solution	Group Contribution (GC)	Quantitative Structure-Property Relationship (QSPR)	Quantum Mechanics based Method (QM-based)
Genetic Algorithm (GA)[133-138]	• **polymer:** materials for semiconductor encapsulation, 1993~1995[328-330] • **small molecules:** alternative refrigerant, 1995[329] • **solvent:** LL extraction, 2000[345] • **solvent:** LL extraction, 2007[346] • **solvent:** antioxidant solubilization, 2014[347]	• **solvent:** LL extraction, 1995[351] • **small molecules (for drug design):** lipophilicity, length, solvent accessible surface, dipole moment, 2000[352] • **polymer (enzyme inhibitor):** ΔG of RNA folding, RNA sequence length, 2002[353]	• **solvent:** LL extraction, 2017[356] • **polymer:** dielectric constant, 2016[331] • **solvent:** reaction rate constant, 2017[357]

doi:10.6342/NTU202403528

	• **solvent:** relative energy difference and solubility parameters, 2014[347] • **solvent:** extractive reaction, 2016[348] • **solvent:** ab/desorption process, 2017[349] • **IL:** heat transfer, 2013[350] • **IL:** electrical conduction, 2013[350] • **IL solvent:** LL extraction, 2013[350] • **IL solvent:** Naphthalene solubilization, 2013[350]	• **small molecules (for drug design):** number of H-bond donors/accepters, docking geometry, 2005[354] • **small molecules:** enzyme-substrate binding energy, structure similarity, 2008[355]	
Simulated Annealing (SA)[140, 358-362]	• **small molecules:** log(K_{ow}), 1996[363] • **small molecules:** alternative refrigerants, 1998[364] • solvent: LL extraction, 1998[364] • **small molecules:** alternative refrigerants, 1998[365]	• **small molecules:** molecular compactness, 1996[363]	

doi:10.6342/NTU202403528

	• **solvent:** LL extraction, 1998[365] • **solvent:** LL extraction, 2002[366, 367] **HSTA** • **solvent:** LL extraction, 2006[368]		
Genetic Algorithm & Simulated Annealing (GA-SA)[141]	• **solvent:** LL extraction, 2017[369]		• **IL:** LL extraction, 2017[370] • **IL:** LL extraction, 2017[371]
Ant Colony Optimization Algorithm (ACO)[146, 372]	• **solvent:** LL extraction, 2015[373] **EACO**		
Tabu Search (TS)[148]		• **metal-ligand complex:** electronegativity, density, toxicity and oxidation state, 2005[374] • **IL:** gas refrigerant separation, 2010[375]	

doi:10.6342/NTU202403528

Solver Package	• **polymer:** mechanical strength, 1996[376] **GINO** • **IL:** azeotrope separation, 2012[377] **GAMS/CPLEX** • **IL:** gas refrigerant separation, 2010[375] **GAMS/CPLEX**	• **metal-ligand complex:** electronegativity, density, toxicity and oxidation state, 2005[374] **GAMS/DICOPT** • **polymer:** glass transition temperature, density and heat capacity, 1999[378] **GAMS/DICOPT++**	
Outer Approximation (OA)[157, 158, 379]	• **solvent:** extractive reaction, 2002[380] • **solvent:** LL extraction, 2002[380] • **solvent:** CO_2 absorption process, 2016[379] • **small molecules:** alternative refrigerants, 1996[381]		
Interval-based Global Optimization Algorithm	• **polymer:** mechanical strength, 1996[376]		

doi:10.6342/NTU202403528

(IBGO)[382]			
Branch-and-Reduce Algorithm (B&R)[383]	• **small molecules:** alternative refrigerants, 2003[384] • **solvent:** LL extraction, 2013[385] • **solvent:** crystallization, 2013[385]		
Brute Force with Reduced Combinatorial Complexity (BF)	• **solvent:** LL extraction, 1983[38] • **solvent:** LL extraction, 1986[386] • **solvent:** LL extraction, 1991[387] • **solvent:** gas absorption, 1991[387] • **solvent:** extractive distillation, 1991[387] • **solvent:** extractive distillation process, 1994[388] • **solvents:** extraction, 1999[389]		

doi:10.6342/NTU202403528

	• **solvent:** extractive distillation, 1999[389] • **solvents:** extraction process, 1999[390] • **solvent:** crystallization, 2006[391] • **IL:** applications on heat transfer, 2013[350] • **IL:** electrical conduction, 2013[350] • **IL solvent:** LL extraction, 2013[350] • **IL solvent:** dissolution of Naphthalene, 2013[350] • **polymer:** density and glass transition temperature, 2015[392] • **surfactant:** UV sunscreen, 2015[392] • **solvent:** LL extraction, 2015[392] • **solvent:** extraction, 1989[39]		

doi:10.6342/NTU202403528

	• **small molecules:** alternative refrigerant, 1989[39] • **polymer:** semiconductor encapsulation, 1989[39]		
*the format of the content is (materials: properties or problem, year)[reference]			

doi:10.6342/NTU202403528

Table A4. The attributes of neutral base elements. Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.

ID	Name (SMILES representation)	Bond order	charge
1	[CH0](-)(-)(-)(-)	1 1 1 1	0
2	C(=)(-)(-)	2 1 1	0
3	C(#)(-)	3 1	0
4	C(=)(=)	2 2	0
5	O(-)(-)	1 1	0
6	O(=)	2	0
7	N(-)(-)(-)	1 1 1	0
8	N(=)(-)	2 1	0
9	N(#)	3	0
10	O(-)	1	0
11	F(-)	1	0
12	Cl(-)	1	0
13	Br(-)	1	0
14	I(-)	1	0
19	S(-)(-)	1 1	0
20	S(=)	2	0
21	P(-)(-)(-)	1 1 1	0
22	P(=)(-)	2 1	0
23	P(#)	3	0
31	[PH0](-)(-)(-)(-)(-)	1 1 1 1 1	0
32	[PH0](=)(-)(-)(-)	2 1 1 1	0
34	S(=)(-)(-)	1 1	0

doi:10.6342/NTU202403528

61	[SH0](=)(=)(-)(-)	2 2 1 1	0
62	Cl(=)(=)(=)(-)	2 2 2 1	0
66	P(=)(-)(-)	2 1 1	0
67	[CH0@@](-)(-)(-)(-)	1 1 1 1	0
68	[CH0@](-)(-)(-)(-)	1 1 1 1	0
69	[PH0](-)(-)(-)(-)	1 1 1 1	0
70	*(-)	1	0

doi:10.6342/NTU202403528

Table A5. The attributes of cation base elements. Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.

ID	Name (SMILES representation)	Bond order	charge
15	[NH0+](-)(-)(-)(-)	1 1 1 1	+1
16	[NH0+](=)(-)(-)	2 1 1	+1
17	[PH0+](-)(-)(-)(-)	1 1 1 1	+1
18	[PH0+](=)(-)(-)	2 1 1	+1
36	[CH0](-)(-)(-)([N+]1C=CN(C)C=1)	1 1 1	+1
37	[CH0](-)(-)(-)([N+]1C=CN(C)C(C)=1)	1 1 1	+1
38	[CH0](-)(-)(-)(N1C=C[N+](C)=C1)	1 1 1	+1
39	[CH0](-)(-)(-)([N+]1=CC=CC(C)=C1)	1 1 1	+1
40	[CH0](-)(-)(-)(C1=C[N+](C)=CC=C1)	1 1 1	+1
41	C(-)(1=[NH+]C=CC=C1)	1	+1
42	[NH0+](-)(1=CC=CC=C1)	1	+1
57	[In+3](-)(-)(-)(-)	1 1 1 1	+3
64	[Ga+3](-)(-)(-)(-)	1 1 1 1	+3
65	[SH0+](-)(-)(-)	1 1 1	+1

doi:10.6342/NTU202403528

Table A6. The attributes of anion base elements. Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.

ID	Name (SMILES representation)	Bond order	charge
24	[F-]	0	-1
25	[Cl-]	0	-1
26	[Br-]	0	-1
27	[I-]	0	-1
28	[OH1-]	0	-1
29	[OH0-](-)	1	-1
30	[PH0-](-)(-)(-)(-)(-)(-)	1 1 1 1 1 1	-1
33	S(-)(=O)(=O)([O-])	1	-1
35	[NH0-](-)(-)	1 1	-1
43	[NH0-](S(=O)(=O)C(F)(F)(F))(S(=O)(=O)C(F)(F)(F))	0	-1
44	[BH0-](-)(-)(-)(-)	1 1 1 1	-1
45	C(-)(-)(-)(C(=O)([O-]))	1 1 1	-1
46	C(#N)([S-])	0	-1
47	C(-)(-)(-)(OP(=O)(OC)([O-]))	1 1 1	-1
48	C(#N)([N-]C#N)	0	-1
49	[BH0-](C#N)(C#N)(C#N)(C#N)	0	-1
50	S(OCCOCCOC)(=O)(=O)([O-])	0	-1
51	S(c(cc1)ccc1C)(=O)(=O)([O-])	0	-1
52	[PH0-](F)(F)(F)(C(C(F)(F)F)(F)F)(C(C(F)(F)F)(F)F)(C(C(F)(F)F)(F)F)	0	-1
53	[NH0-](S(=O)(=O)C(C(F)(F)F)(F)F)(S(=O)(=O)C(C(F)(F)F)(F)F)	0	-1

doi:10.6342/NTU202403528

54	[CH0-](S(C(F)(F)(F))(=O)(=O))(S(C(F)(F)(F))(=O)(=O))(S(C(F)(F)(F))(=O)(=O))	0	-1
55	[PH0-](F)(F)(F)(F)(F)(F)	0	-1
56	[In+3]([Cl-])([Cl-])([Cl-])([Cl-])	0	-1
58	Cl(=O)(=O)([O-])(=O)	0	-1
59	[CH0-](-)(-)(-)	1 1 1	-1
60	[NH0+](=O)([O-])([O-])	0	-1
63	[SH0-](-)	1	-1

doi:10.6342/NTU202403528

Algorithm 1. A code segment for converting SMILES into MARS+ MDS format. Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.

MDS consists of element index array (Cindex), parent index array (Pindex), element type array (Mindex), bond order array (Rindex), cyclic flag array (Cyindex), cyclic bond array (Cybnd), cis-tran front/end flag array (ctsisomer), and chirality flag array (chi). Now we describe how to convert a SMILES into MDS with the aid of OpenBabel.

1. Openbabel: Convert a given SMILES to an OBMol object

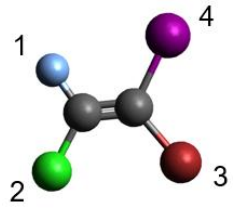
```
OBMol mol; //Declare mol as an instance of OBMol object
OBConversion conv(&SMILES_stringstream); //Input the given SMILES to
conv
conv.SetInFormat("SMI"); //Set SMILES to be the input format for conv
conv.Read(&mol); //Use conv to build the connectivity among heavy atoms
in mol
mol.AddHydrogens(); //Add hydrogen atoms to the heavy atoms in mol
PerceiveStereo(&mol); //Perceive chiral centers from current mol
OBBuilder builder;
builder.Build(mol); //Create 3D coordinates for all the atoms in mol
PerceiveStereo(&mol); //Perceive chiral centers from 3D structure of
mol
```

2. Transcribe the structural information (atoms, bonding, and isomerisms) in mol to MDS

```
OBStereoFacade facade(&mol); //Use facade to get isomerism information
of mol
FOR_BONDS_OF_MOL(bond, mol) { //Traverse all the bonds in mol
   //Access the two atoms connected by this bond
   OBAtom* BeginAtom=bond->GetBeginAtom();
   OBAtom* EndAtom=bond->GetEndAtom();
   //Transcribe the information to bond order array (Rindex)
   Rindex.at(EndAtom->GetId())=bond->GetBondOrder();

   //Perceive cis-trans isomerisms in mol
   if   (bond->GetBondOrder()==2   &&   facade.HasCisTransStereo(bond-
>GetId())) {
      OBCisTransStereo* ct=NULL;
      ct=facade.GetCisTransStereo(bond->GetId());
      OBTetrahedralStereo::Config A=tr->GetConfig(OBStereo::ShapeU);
      //Access the substituents in U-shape order (1234, 2341, 3412, or
4123):
       A.refs.at(0), A.refs.at(1), A.refs.at(2), and A.refs.at(3)
```


doi:10.6342/NTU202403528

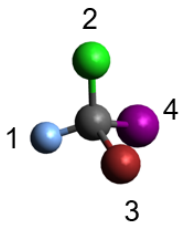


SMILES: F(/C(=C/(Br)(I))(Cl))

```
      //It does not need to know what permutation is actually adopted.
      //Generally, the 1st and 3rd accessed atoms are in trans
        configuration.
      //Assign "/" as the cis-trans front flag (Fct) for the beginning
atom (C)
        Of the double bond.
      //Assign "/" as the cis-trans end flag (Ect) for the ending atom
(C) of
        the double bond.
      OBAtom* BeginAtom=bond->GetBeginAtom();
      OBAtom* EndAtom=bond->GetEndAtom();
      ctsisomer.at(0).at(BeginAtom->GetId())="/";    //Cis-trans front
flag
      ctsisomer.at(1).at(EndAtom->GetId())="/";  //Cis-trans end flag
   }
}

FOR_ATOMS_OF_MOL(atom, mol) { //Traverse all the atoms in mol
   //Transcribe information to element index array (Cindex)
   Cindex.at(atom->GetId())=atom->GetId()+1;
   if (atom->GetAtomNum()==6) {
      //Transcribe information to element type array (Mindex), based on
valences.
      if (the atom has 4 single bonds) Mindex.at(atom->GetId())=1;
      if (the atom has 2 single bonds and 1 double bond) ...
   }
   else if (atom->GetAtomNum()==7) {
       ...
   }

   //Perceive optical isomerisms in mol
   if (facade.HasTetrahedralStereo(atom->GetId())) {
      OBTetrahedralStereo* tr=NULL;
      tr=facade.GetTetrahedralStereo(atom->GetId());
      OBTetrahedralStereo::Config                                  A=tr-
>GetConfig(OBStereo::Clockwise);
      //Access the substituents in clockwise order (234, 342, or 423):
        A.refs.at(0), A.refs.at(1), and A.refs.at(2)
```



SMILES: F([C@@](Cl)(Br)(I))

doi:10.6342/NTU202403528

```
      //Access the chiral center: A.center
      //Check if the order of these substituents in mol are consistent
with
        the clockwise order.
      If (consistent order) chi.at(A.center)=2;  //Chirality flag
      else chi.at(A.center)=1;  //Chirality flag
   }
}
unsigned int ringnum=1;
FOR_BONDS_OF_MOL(bond, mol) { //Traverse all the bonds in mol
   //Assign parent indices
   OBAtom* BeginAtom=bond->GetBeginAtom();
   OBAtom* EndAtom=bond->GetEndAtom();
   if (BeginAtom is the 1st parent of EndAtom) {
      //Record the numbering of BeginAtom in parent indices array
      Pindex.at(EndAtom->GetId())=Cindex.at(BeginAtom->GetId());
   }
   else if (BeginAtom is the 2nd parent of EndAtom) {
      //Record this bonding in cyclic flag array and cyclic bond order
array
      Cyindex.at(BeginAtom->GetId()).push_back(ringnum);
      Cyindex.at(EndAtom->GetId()).push_back(ringnum);
      Cybnd.resize(ringnum,0);
      Cybnd.at(ringnum-1)=bond->GetBondOrder();
      ringnum++;
   }
}
```


doi:10.6342/NTU202403528

Algorithm 2. Steps to convert MARS+ MDS into SMILES. Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.

Suppose we have ethane (raw SMILES: C(C)) in MDS format:

Parent indices (Pindex)	0 1
Element indices (Cindex)	1 2
Element types (Mindex)	1 1
Bond orders (Rindex)	0 1
Cyclic flag array (Cyindex)	0 0
Cyclic bond order array (Cybnd)	Null
Cis-trans front flag array (ctsisomer.at(0))	_ _
Cis-trans end flag array (ctsisomer.at(1))	_ _
Chirality flag array	0 0

1. Create *Bindex* and *atomsmi* for the molecule

The first C atom:

```
//See main text sec 2.1 for the meaning of name, index, and suffspos.
```

name: C**(-)**(-)(-)(-)

index=2

suffspos=13

```
//Bindex[i] records the conectivity of the (i+1)th element
//Its 1st bond is used to connect with the 2nd C atom.
//3 remaining single bonds are free (connect with H atoms)
```

Bindex[0] = [**0**, 1, 1, 1]

```
//For the (i+1)th element in the molecule, atomsmi[i] records the output
   positions for its name.
//Initialization of output positions for each of the charaters in its name.
```

atomsmi[0] = [0, **1**, **2**, **3**, 4, 5, 6, 7, 8, 9, 10, 11, 12]

The second C atom:

name: C**(-)**(-)(-)(-)

index=2

suffspos =13

```
//Its 1st bond is used to connect with the 1st C atom
```

Bindex[1] = [**0**, 1, 1, 1]

```
//Initialization of output positions for each of the charaters in its name.
```

atomsmi[1] = [0, **1**, **2**, **3**, 4, 5, 6, 7, 8, 9, 10, 11, 12]

2. MDS: Shift the output position of characters

doi:10.6342/NTU202403528

//Adjust atomsmi[0] and atomsmi[1] so that the *name* of the two atoms can be outputted as C(-C(-)(-)(-)(-))(-)(-)(-), where the blue characters come from the 1st C atom, and the brown characters come from the 2nd C atom

The first C: For each character after position $index_{1st\_C}$, shift the output position by $nbond_{2nd\_C}$

name: C**(-)**(-)(-)(-)

index=2

suffspos =13

Bindex[0] = [**0**, 1, 1, 1]

atomsmi[0] = [0, **1**, **2**, **16**, 17, 18, 19, 20, 21, 22, 23, 24, 25] //Adjusted

The second C: Shift the output position by (1+$index_{1st\_C}$) for all the characters.

name: C**(-)**(-)(-)(-)

index=2

suffspos =13

Bindex[1] = [**0**, 1, 1, 1]

atomsmi[1] = [3, **4**, **5**, **6**, 7, 8, 9, 10, 11, 12, 13, 14, 15] //Adjusted

3. MDS: Erase redundant "(-)" and "-".

//With the aid of Bindex[0] and Bindex[1], one can erase the redundant notations for single bonds "-" and free valences "(-)". As a result, the SMILES would be C(_C_____)_____, where the underline denotes white spaces.

The first C:

name: C**()**

Bindex[0] = [**0**, 1, 1, 1]

atomsmi[0] = [0, **1**, **16**] //Redundant bond notations erased.

The second C:

name: C

Bindex[1] = [0, 1, 1, 1]

atomsmi[1] = [3] //Redundant bond notations erased.

4. MDS: Re-numbers the output position of the remaining characters to form a SMILES string.

//Now delete the white spaces of C(_C_____)_____. Arrange consecutive output positions for the remaining characters so that reasonable SMILES C(C) can be generated.

The first C:

name: C**()**

Bindex[0] = [**0**, 1, 1, 1]

atomsmi[0] = [0, **1**, **3**] //Renumbered.

The second C:

doi:10.6342/NTU202403528

name: C

Bindex[1] = [0, 1, 1, 1]

atomsmi[1] = [2] `//Renumbered.`

5. MDS: Output raw SMILES by writing the character *name[i][j]* to the *atom[i][j]* position in a character array *Rawsmi*

Rawsmi[atomsmi[0][0]]=Rawsmi[0]="C"

Rawsmi[atomsmi[0][1]]=Rawsmi[1]="("

Rawsmi[atomsmi[1][0]]=Rawsmi[2]="C"

Rawsmi[atomsmi[0][3]]=Rawsmi[3]=")"

→ *Rawsmi* = C(C)

6. OpenBabel: Canonicalize the raw SMILES

```
//Specify Rawsmi in RAW_SMILES_stringstream
//The canonical SMILES is outputted to CAN_SMILES_stringstream
OBConversion conv(&RAW_SMILES_stringstream,&CAN_SMILES_stringstream);

//Canonicalize the raw SMILES
if(conv.SetInAndOutFormats("SMI","SMI")) {
    conv.AddOption("canonical",OBConversion::GENOPTIONS);
    conv.Convert();
}
```


doi:10.6342/NTU202403528

Appendix B. Supplementary Figures to the Main Texts

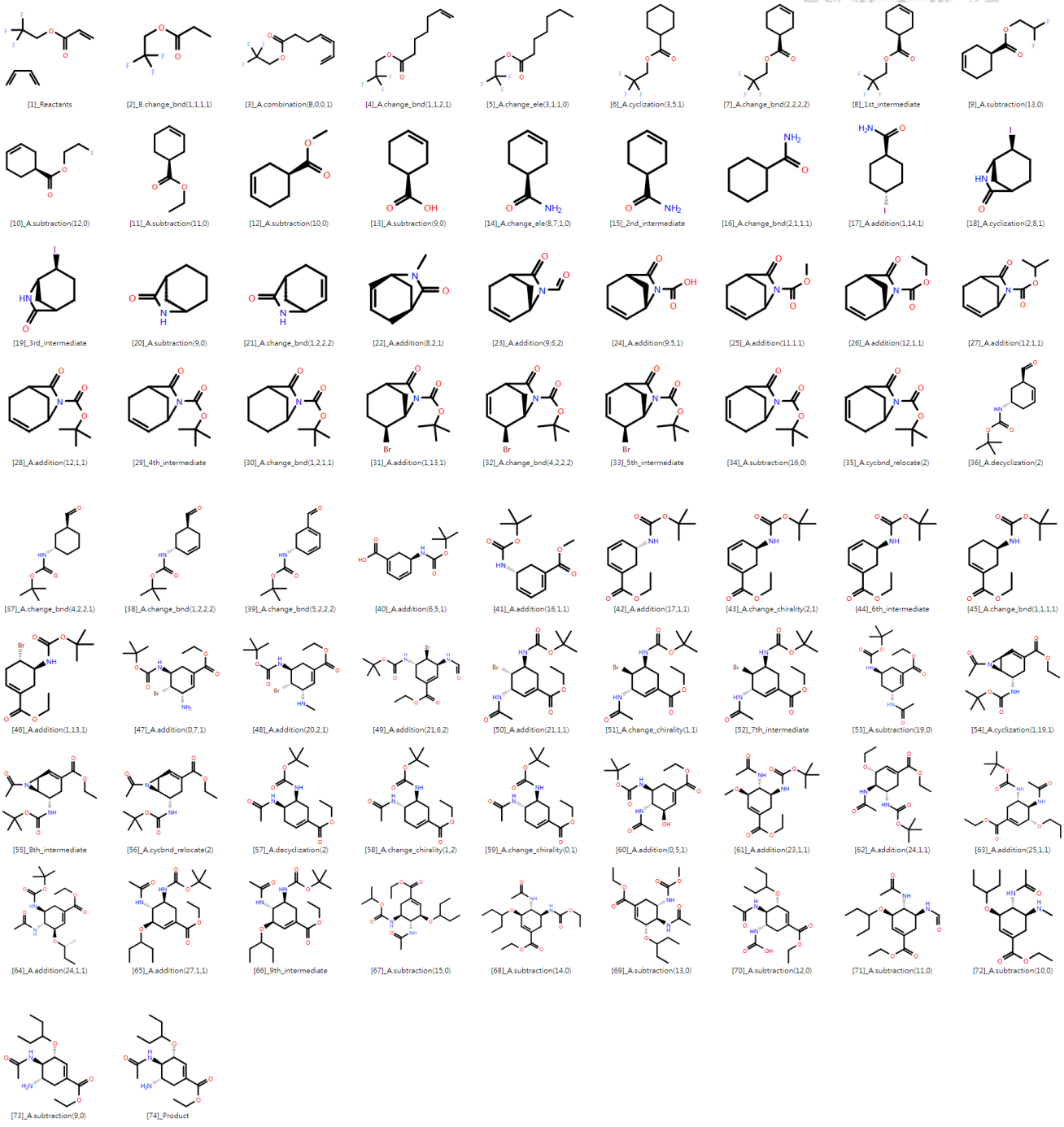


Figure B1. Constructing a programmatic sequence of molecular operations to mimic the Oseltamivir synthesis pathway of E.J. Corey et al. The caption under a structure indicates the operation to bring the previous structure to current one. Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.

doi:10.6342/NTU202403528

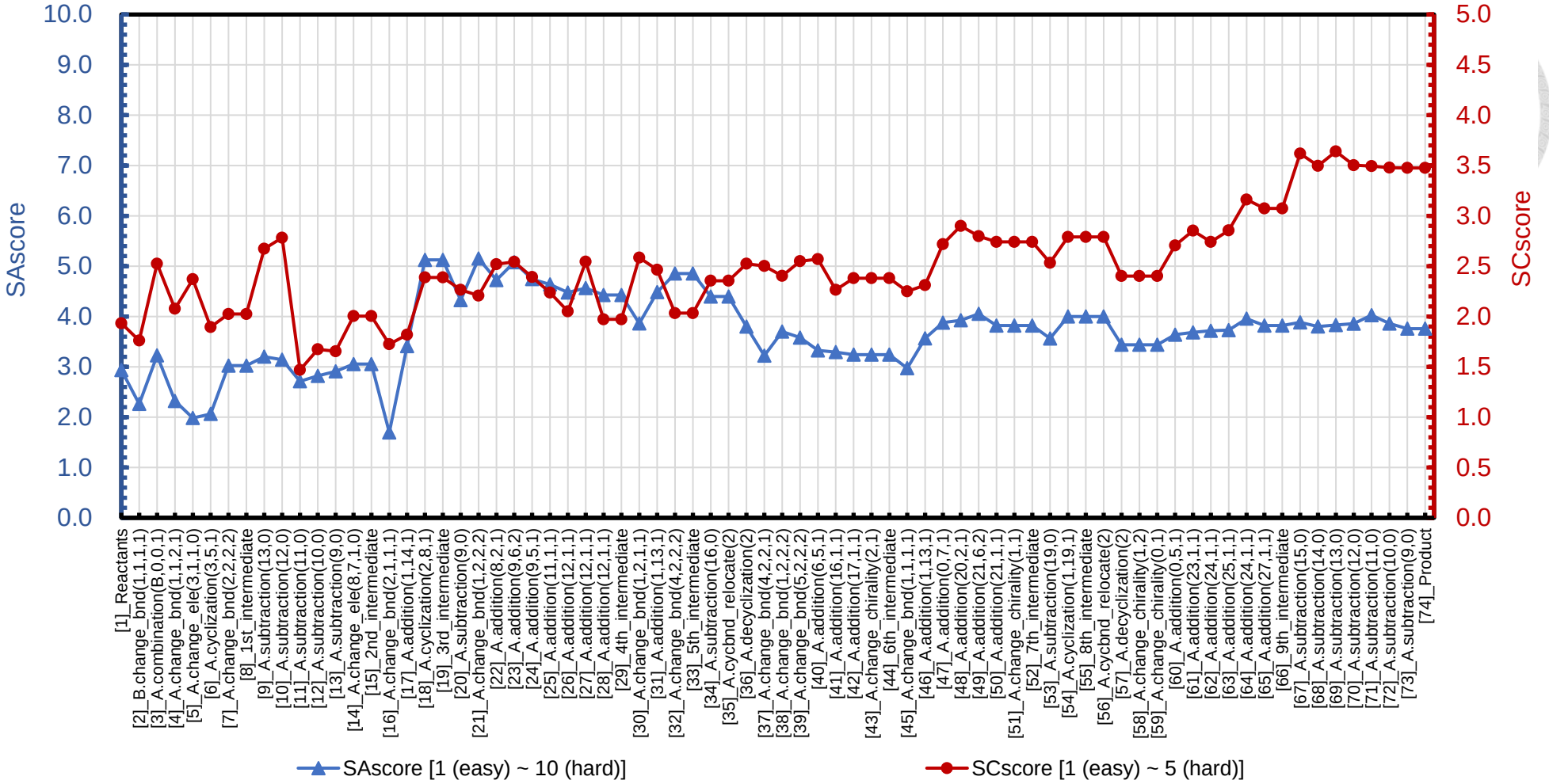


Figure B2. Constructing a programmatic sequence of molecular operations to mimic the Oseltamivir synthesis pathway of E.J. Corey et al.: the variation of SCscore and SAscore with respect to reaction steps. Reprinted with permission from the reference[163]. Copyright 2023 American Chemical Society.

doi:10.6342/NTU202403528

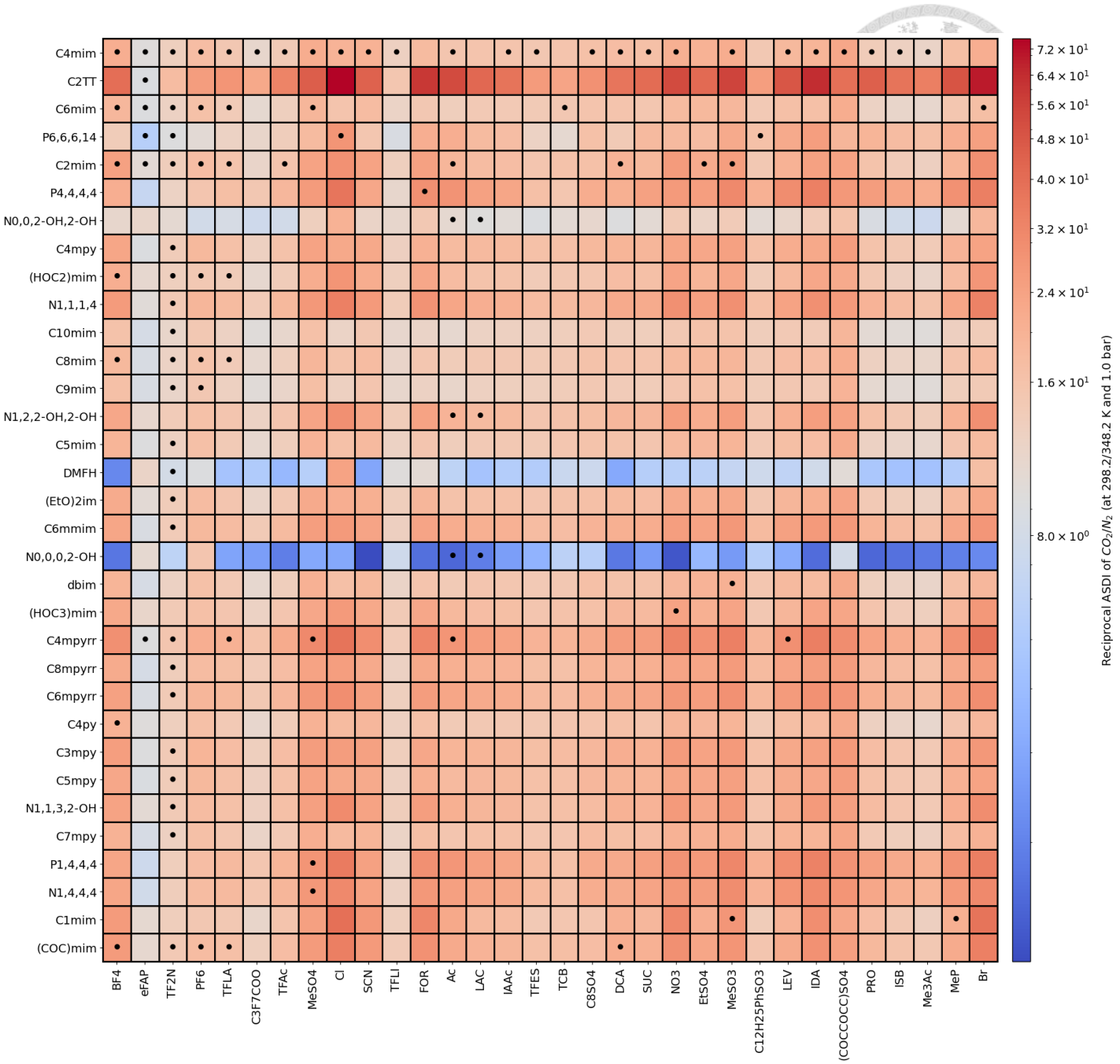


Figure B3. Reciprocal absorption-selectivity-desorption index (ASDI, see eq (5.2–11)) of CO_2 over N2 in the screened ILs. The darker red indicates the desirable performance. A dot in a cell indicates the presence of at least one VLE experimental data point for the CO_2-IL system.

doi:10.6342/NTU202403528

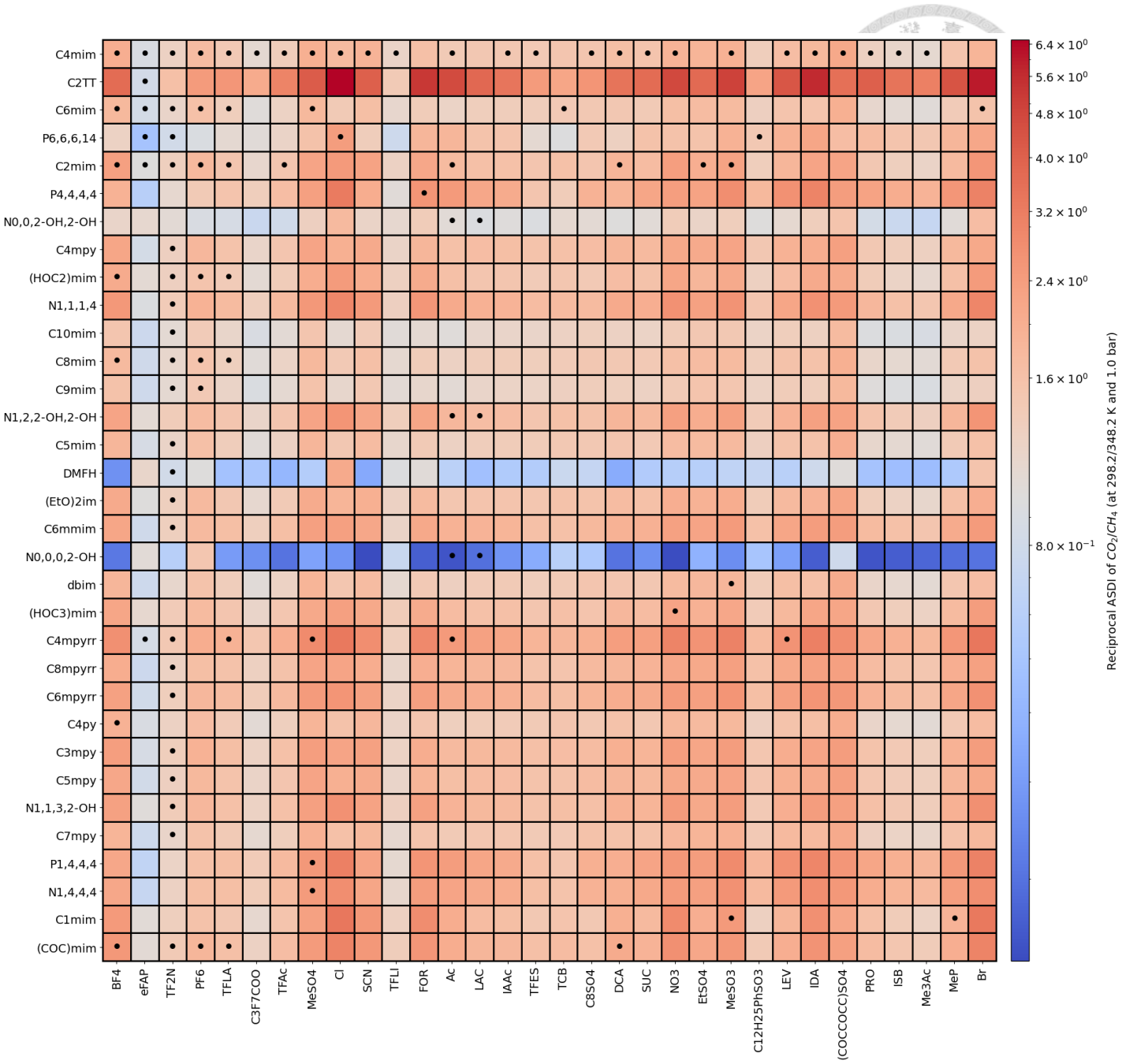


Figure B4. Reciprocal absorption-selectivity-desorption index (ASDI, see eq (5.2–11)) of CO_2 over CH4 in the screened ILs. The darker red indicates the desirable performance. A dot in a cell indicates the presence of at least one VLE experimental data point for the CO_2-IL system.

doi:10.6342/NTU202403528

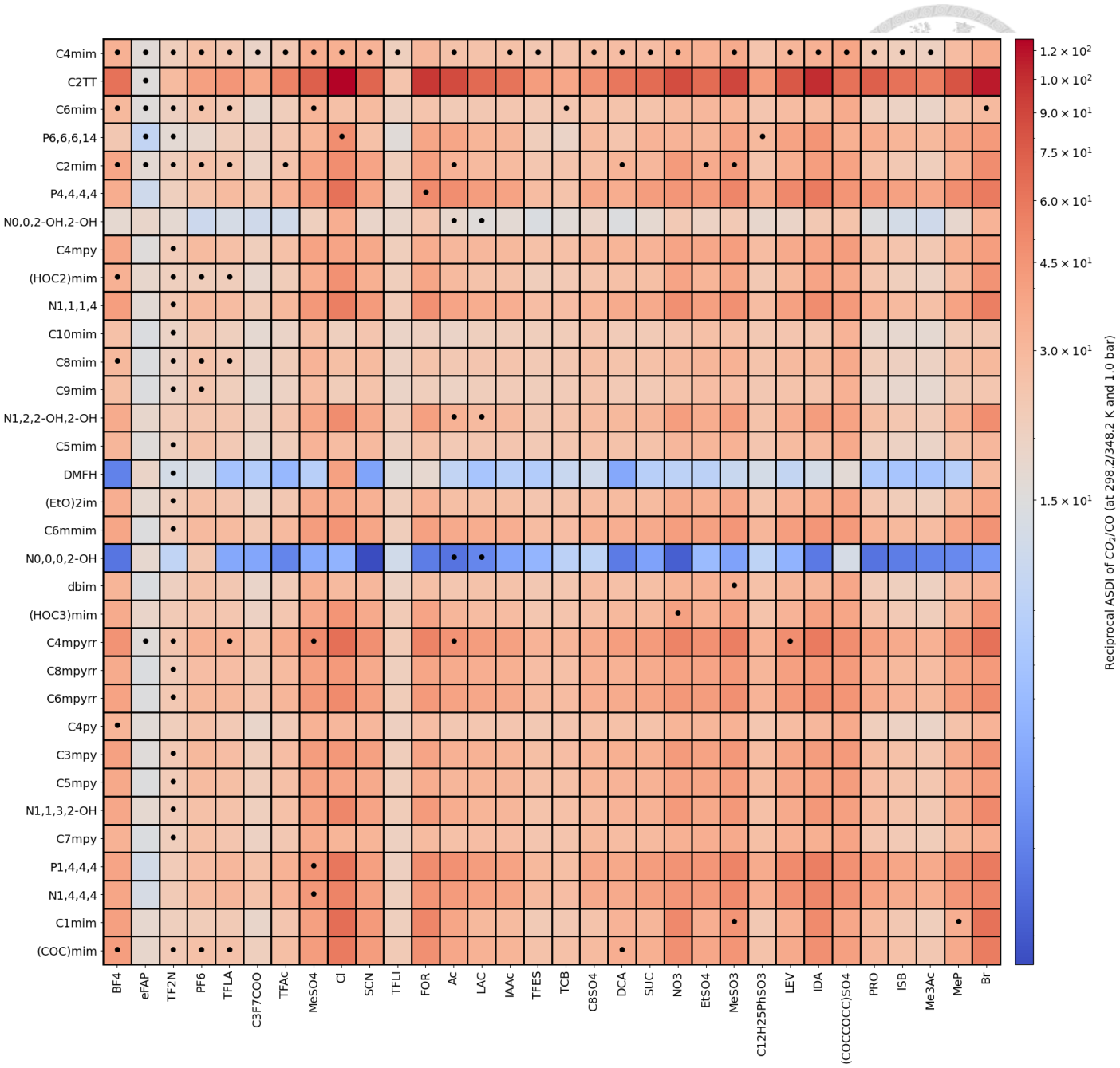


Figure B5. Reciprocal absorption-selectivity-desorption index (ASDI, see eq (5.2–11)) of CO_2 over CO in the screened ILs. The darker red indicates the desirable performance. A dot in a cell indicates the presence of at least one VLE experimental data point for the CO_2-IL system.

doi:10.6342/NTU202403528

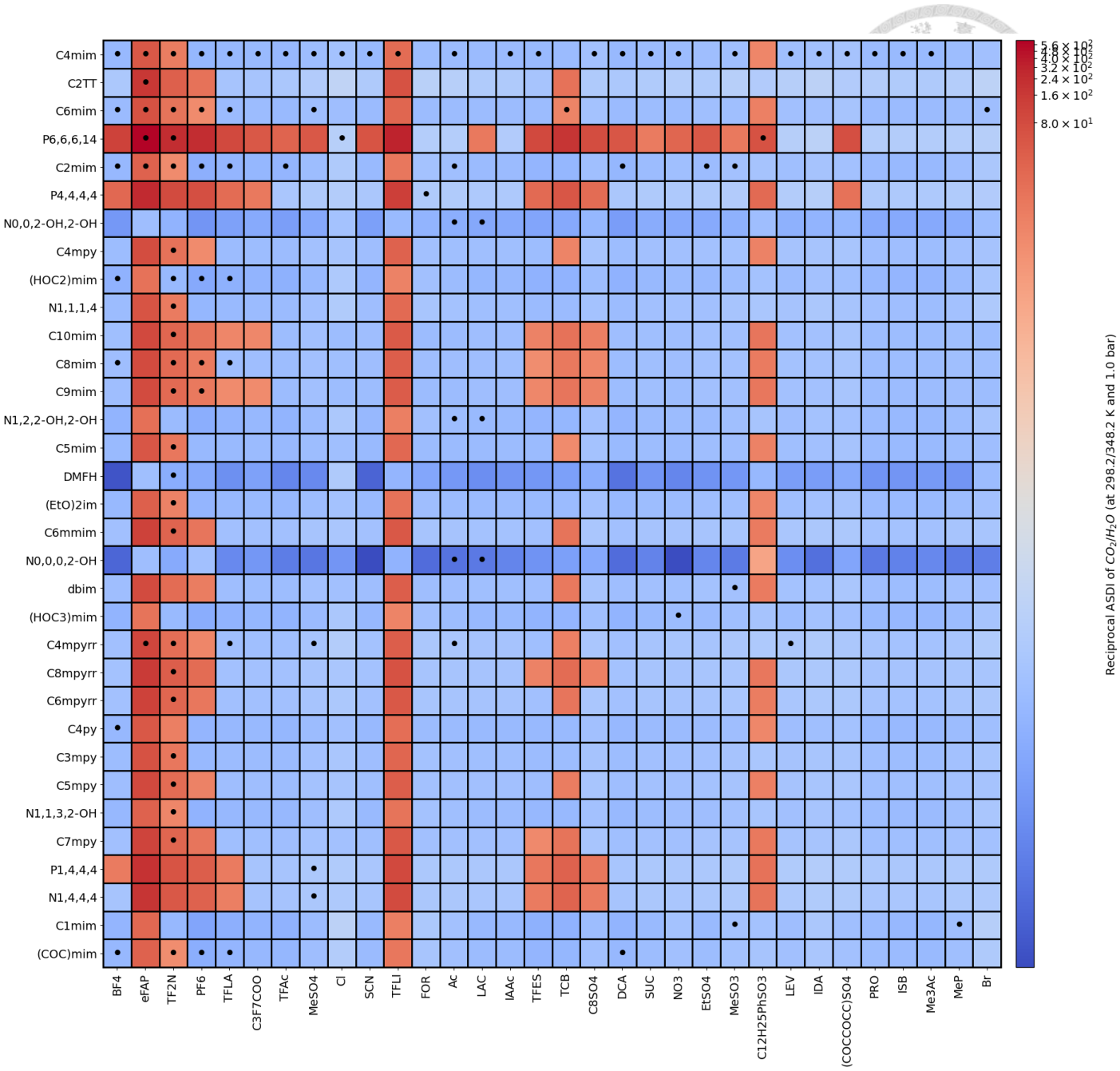


Figure B6. Reciprocal absorption-selectivity-desorption index (ASDI, see eq (5.2–11)) of CO_2 over H_2O in the screened ILs. The darker red indicates the desirable performance. A dot in a cell indicates the presence of at least one VLE experimental data point for the CO_2-IL system.

doi:10.6342/NTU202403528

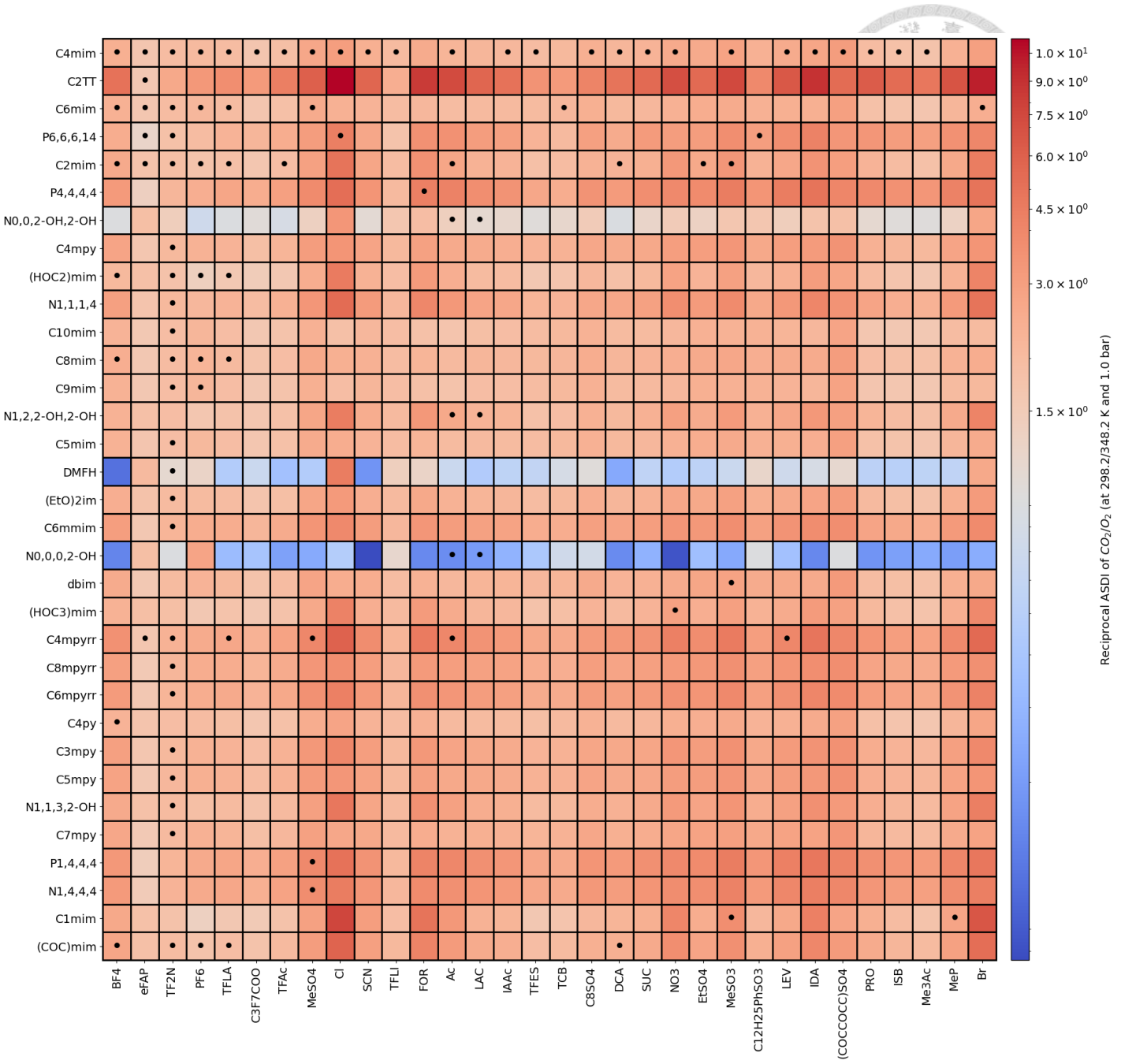


Figure B7. Reciprocal absorption-selectivity-desorption index (ASDI, see eq (5.2–11)) of CO_2 over O_2 in the screened ILs. The darker red indicates the desirable performance. A dot in a cell indicates the presence of at least one VLE experimental data point for the CO_2-IL system.

doi:10.6342/NTU202403528

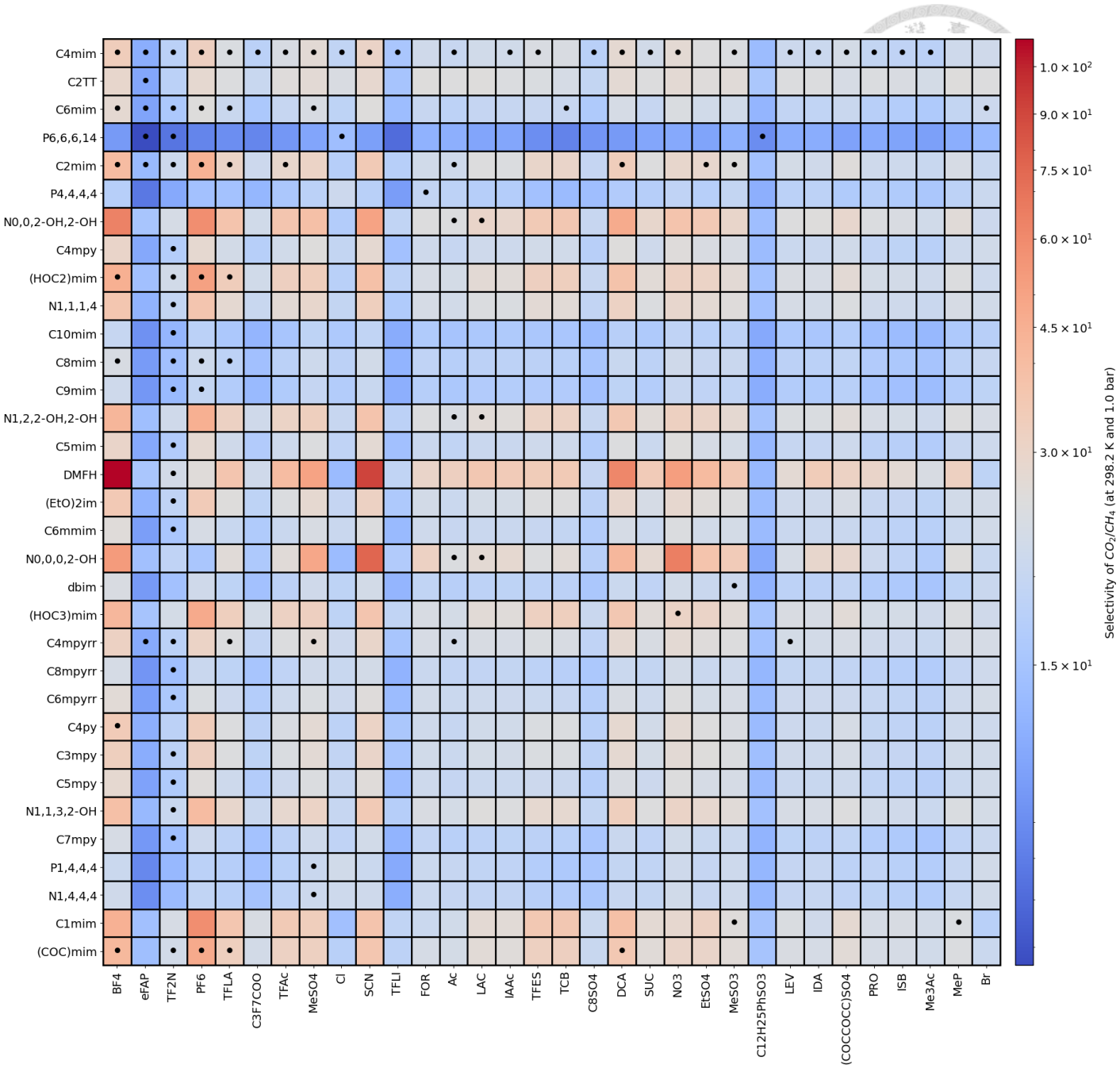


Figure B8. Selectivity of CO_2 over CH_4 (eq (5.2–8)) in the screened ILs. The darker red indicates the desirable performance. A dot in a cell indicates the presence of at least one VLE experimental data point for the CO_2-IL system.

doi:10.6342/NTU202403528

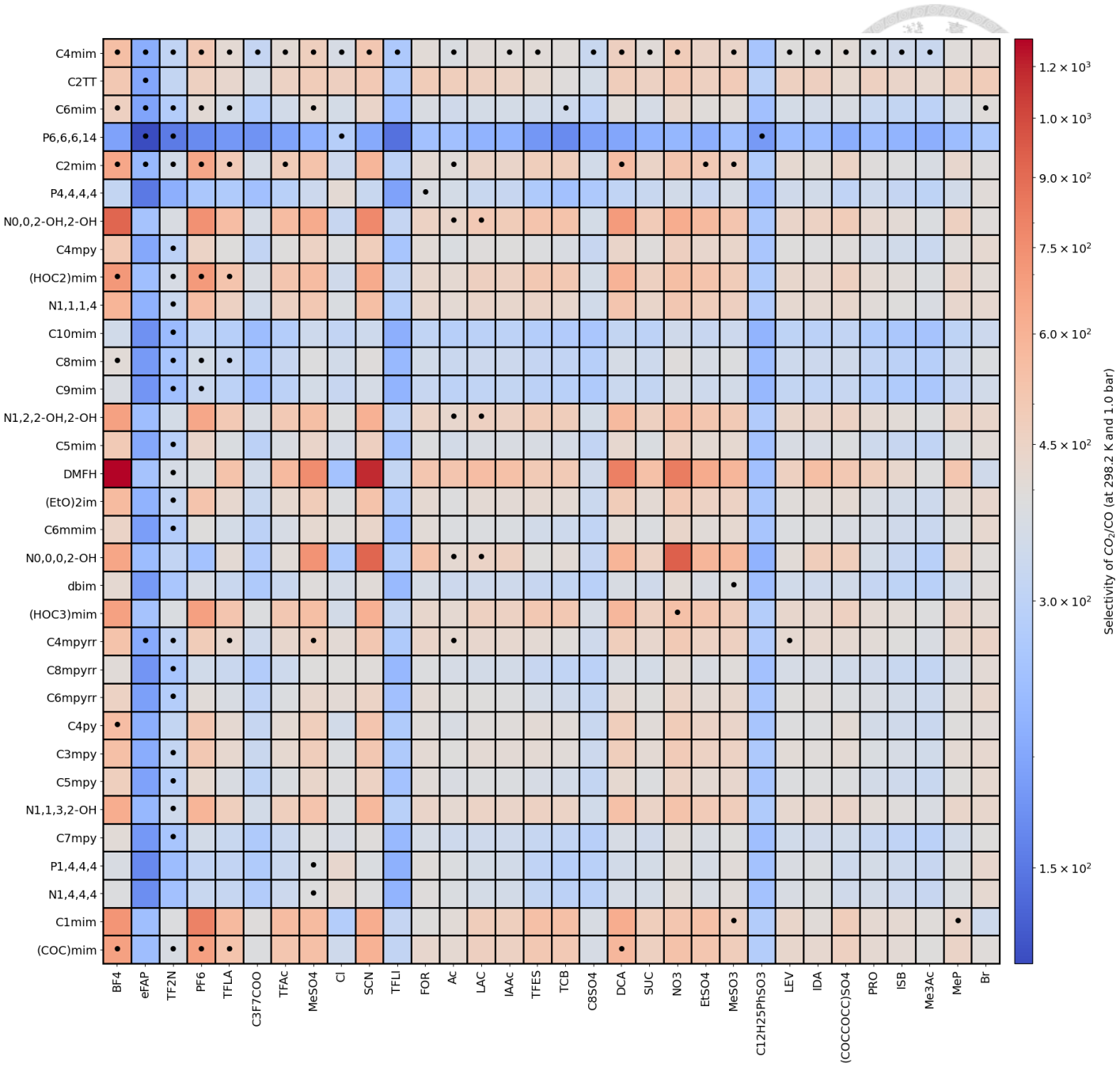


Figure B9. Selectivity of CO_2 over CO (eq (5.2–8)) in the screened ILs. The darker red indicates the desirable performance. A dot in a cell indicates the presence of at least one VLE experimental data point for the CO_2-IL system.

doi:10.6342/NTU202403528

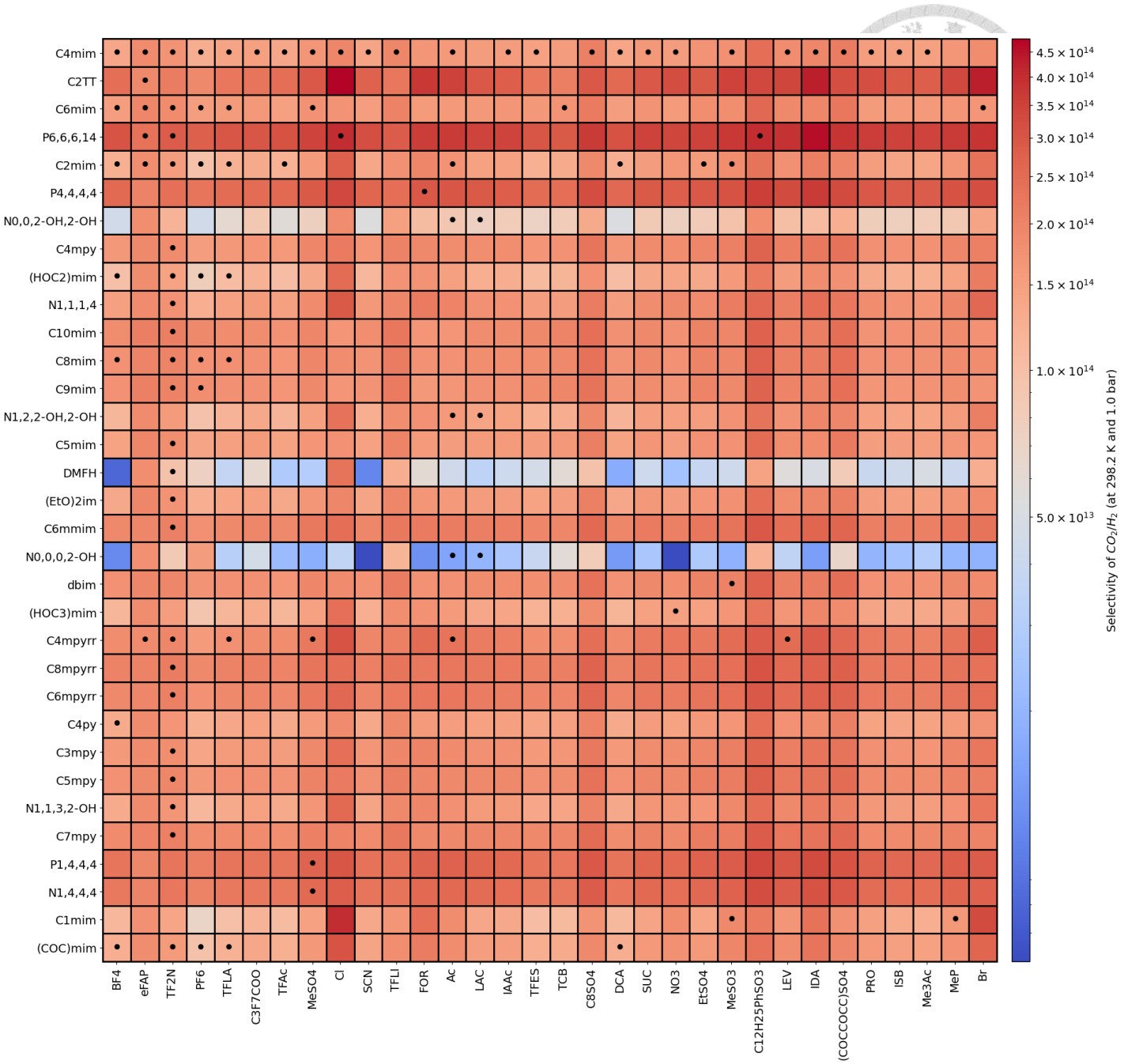


Figure B10. Selectivity of CO_2 over H_2 (eq (5.2–8)) in the screened ILs. The darker red indicates the desirable performance. A dot in a cell indicates the presence of at least one VLE experimental data point for the CO_2-IL system.

doi:10.6342/NTU202403528

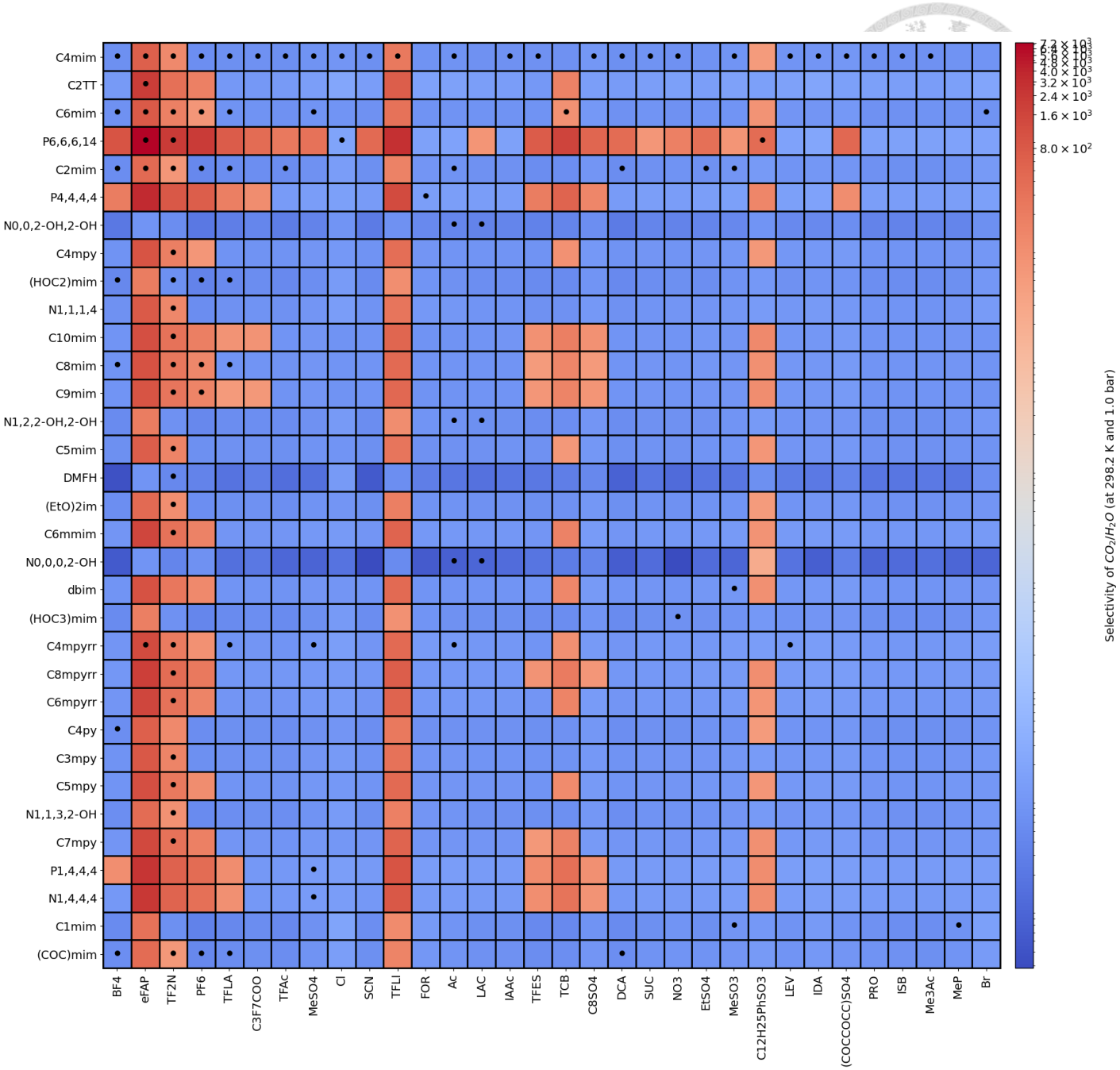


Figure B11. Selectivity of CO_2 over H_2O (eq (5.2–8)) in the screened ILs. The darker red indicates the desirable performance. A dot in a cell indicates the presence of at least one VLE experimental data point for the CO_2-IL system.

doi:10.6342/NTU202403528

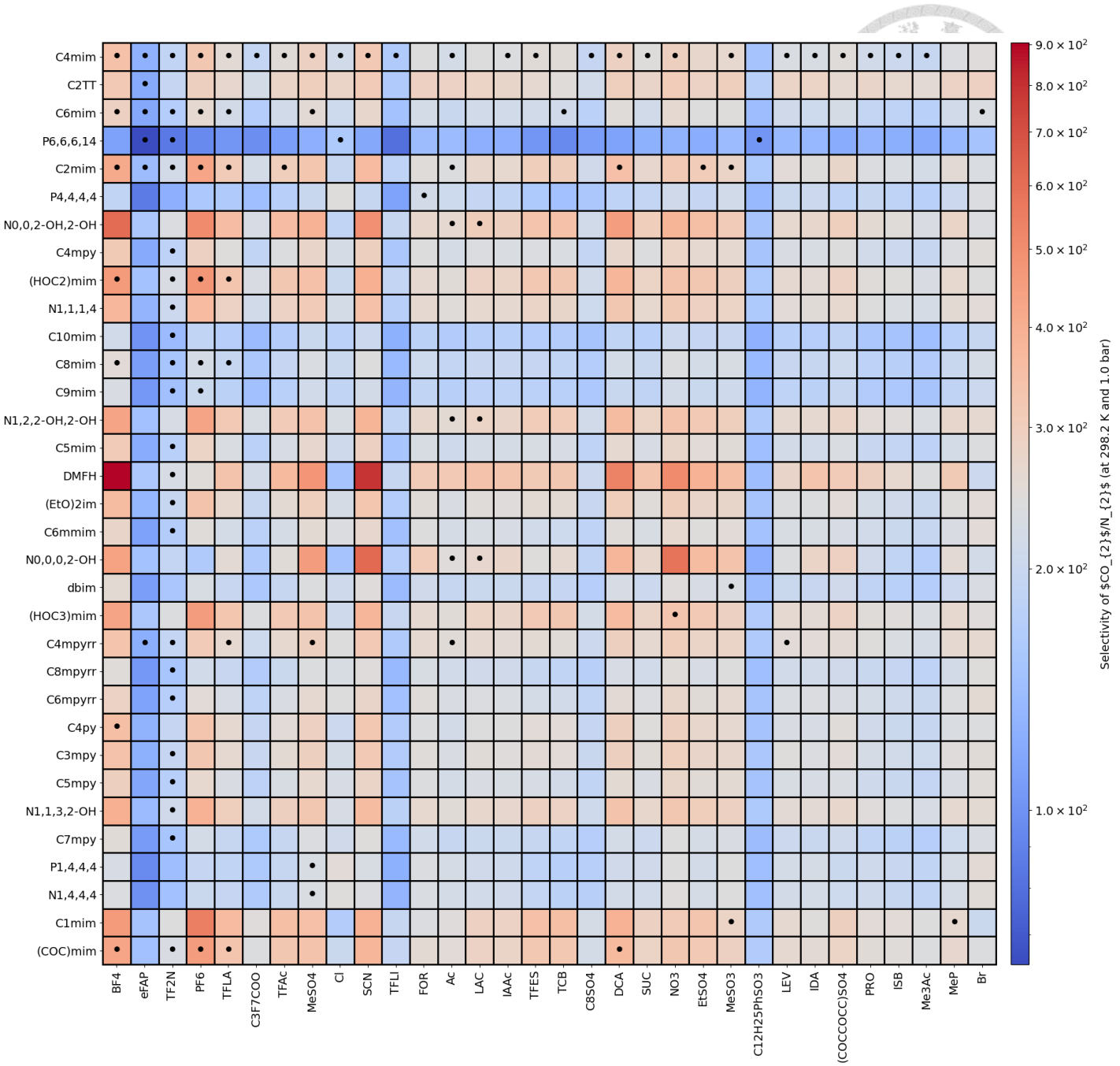


Figure B12. Selectivity of CO_2 over N_2 (eq (5.2–8)) in the screened ILs. The darker red indicates the desirable performance. A dot in a cell indicates the presence of at least one VLE experimental data point for the CO_2-IL system.

doi:10.6342/NTU202403528

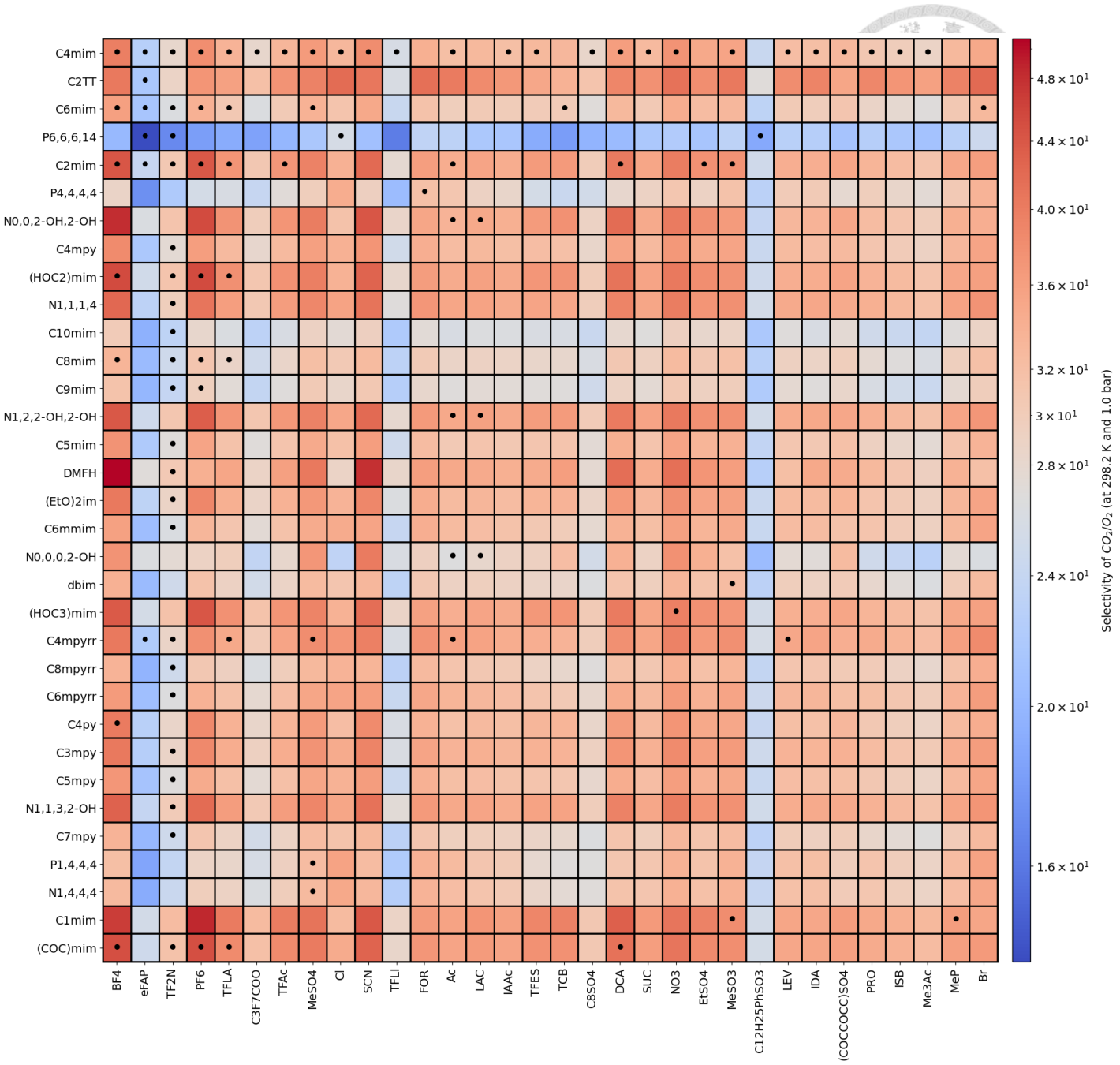


Figure B13. Selectivity of CO_2 over O_2 (eq (5.2–8)) in the screened ILs. The darker red indicates the desirable performance. A dot in a cell indicates the presence of at least one VLE experimental data point for the CO_2-IL system.

doi:10.6342/NTU202403528

Appendix C. Optimality in Non-linear Programming

C.1. Optimality Conditions in Unconstrained Optimizations[393]

Problem C1

$$\underset{\boldsymbol{w}}{\operatorname{argmin}}\ Objfcn(\boldsymbol{w}) \tag{C.1–1}$$

$$\boldsymbol{w} \in W \subseteq \mathbb{R}^m \tag{C.1–2}$$

C.1.1. First-order Necessary Conditions

Let $Objfcn(\boldsymbol{w})$ be at least once continuously differentiable in the neighborhood of a point $\boldsymbol{w}^*$. If $\boldsymbol{w}^*$ is a local minimum solution of $Objfcn(\boldsymbol{w})$, then

$$\boldsymbol{\nabla}_{\mathbf{w}} Objfcn(\boldsymbol{w}^*) = \mathbf{0} \tag{C.1–3}$$

C.1.2. Second-order Necessary Conditions

Let $Objfcn(\boldsymbol{w})$ be at least twice continuously differentiable in the neighborhood of a point $\boldsymbol{w}^*$. If $\boldsymbol{w}^*$ leads to a local minimum of $Objfcn(\boldsymbol{w})$, then

(I) $$\boldsymbol{\nabla}_{\mathbf{w}} Objfcn(\boldsymbol{w}^*) = \mathbf{0} \tag{C.1–4}$$

(II) $$\boldsymbol{w}^{\boldsymbol{T}}[\boldsymbol{\nabla}_{\mathbf{w}}\boldsymbol{\nabla}_{\mathbf{w}} Objfcn(\boldsymbol{w}^*)]\boldsymbol{w} \geq 0, \forall \boldsymbol{w} \in \mathbb{R}^m \tag{C.1–5}$$

Property (II) means that $[\boldsymbol{\nabla}_{\mathbf{w}}\boldsymbol{\nabla}_{\mathbf{w}} Objfcn(\boldsymbol{w}^*)]$ matrix is positive semidefinite.

C.1.3. Second-order Sufficient Conditions

Let that $Objfcn(\boldsymbol{w})$ is at least twice continuously differentiable in the

doi:10.6342/NTU202403528

neighborhood of a point $\boldsymbol{w}^*$. If

$$\text{(I)}\ \ \boldsymbol{\nabla_w} Objfcn(\boldsymbol{w}^*) = \boldsymbol{0} \tag{C.1–6}$$

$$\text{(II)}\ \ \boldsymbol{w^T}[\boldsymbol{\nabla_w}\boldsymbol{\nabla_w} Objfcn(\boldsymbol{w}^*)]\boldsymbol{w} \geq 0, \forall \boldsymbol{w} \in \mathbb{R}^m \tag{C.1–7}$$

then $\boldsymbol{w}^*$ is a local minimum. Note that if $[\boldsymbol{\nabla_w}\boldsymbol{\nabla_w} Objfcn(\boldsymbol{w}^*)]$ matrix in condition (II) is positive definite (i.e. $\boldsymbol{w^T}[\boldsymbol{\nabla_w}\boldsymbol{\nabla_w} Objfcn(\boldsymbol{w}^*)]\boldsymbol{w} > 0$ for all $\boldsymbol{w} \in \mathbb{R}^m$), then $\boldsymbol{w}^*$ will be a strict local minimum.

C.1.4. A Short Proof

These theorems can be examined by second-order Tyler expansion at local minimum solution $\boldsymbol{w}^*$ along a feasible direction (section C.3).

$$\begin{aligned} &Objfcn(\boldsymbol{w}^* + t\boldsymbol{d}) \\ &\quad = Objfcn(\boldsymbol{w}^*) + t\boldsymbol{d^T}\boldsymbol{\nabla_w} Objfcn(\boldsymbol{w}^*) \\ &\quad + \frac{1}{2}t^2\boldsymbol{d^T}[\boldsymbol{\nabla_w}\boldsymbol{\nabla_w} Objfcn(\boldsymbol{w}^*)]\boldsymbol{d} + \mathcal{O}(t^3) \end{aligned} \tag{C.1–8}$$

C.2. Optimality Conditions in Constrained Optimizations[37, 394]

Problem C2

$$\underset{\boldsymbol{w}}{\text{argmin}}\ Objfcn(\boldsymbol{w}) \tag{C.2–1}$$

subjected to

$$\boldsymbol{h}(\boldsymbol{w}) = 0 \tag{C.2–2}$$

$$\boldsymbol{g}(\boldsymbol{w}) \leq 0 \tag{C.2–3}$$

doi:10.6342/NTU202403528

$$\boldsymbol{w} \in W \subseteq \mathbb{R}^m \tag{C.2–4}$$

C.2.1. First-order Necessary Condition (Karush-Kuhn-Tucker, KKT)

Each of the inequality constraints in eq (C.2–3) can be transformed into an equality one by introducing a slack variable $s_j^2 \geq 0$, that is,

$$g_j(\boldsymbol{w}) + s_j^2 = 0 \tag{C.2–5}$$

After such transformation, the original minimization Problem C2 can be reformulated using Lagrange multiplier method. Let $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_p]^T$ be the multipliers for p equality constraints $\boldsymbol{h}(\boldsymbol{w})$, $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_q]^T$ be the multipliers for q inequality constraints $\boldsymbol{g}(\boldsymbol{w})$, and $\boldsymbol{s} \odot \boldsymbol{s} = [s_1^2, \ldots, s_q^2]^T$ be the slacks variables for each inequality constraints. Here, $\odot$ is the Hardamard product operator, meaning the element-wise product of two matrices with the same dimensions. The Lagrange function $\mathcal{L}(\boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{s})$ is:

$$\mathcal{L}(\boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{s}) = Objfcn(\boldsymbol{w}) + \sum_{i=1}^{p} \lambda_i h_i(\boldsymbol{w}) + \sum_{j=1}^{q} \mu_j [g_j(\boldsymbol{w}) + s_j^2] \tag{C.2–6}$$

Suppose $\boldsymbol{w}^*$ is a local minimum solution to the original Problem C2. There should also be the solution $(\boldsymbol{w}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*, \mathbf{s}^*)$ to the minimization problem in Lagrange multiplier formulation, i.e. eq (C.2–6). For $(\boldsymbol{w}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*, \mathbf{s}^*)$ to be a local extremum, the necessary conditions of eq (C.2–7) to eq (C.2–10) should be satisfied:

doi:10.6342/NTU202403528

$$\nabla_{\mathbf{w}}\mathcal{L}(\boldsymbol{w}^*,\boldsymbol{\lambda}^*,\boldsymbol{\mu}^*,\boldsymbol{s}^*) = \nabla_{\mathbf{w}}Objfcn(\boldsymbol{w}^*) + \sum_{i=1}^{p}\lambda_i^*\nabla_{\mathbf{w}}h_i(\boldsymbol{w}^*) + \sum_{j=1}^{q}\mu_j^*\nabla_{\mathbf{w}}g_j(\boldsymbol{w}^*) = \mathbf{0} \quad \text{(C.2–7)}$$

$$\nabla_{\boldsymbol{\lambda}}\mathcal{L}(\boldsymbol{w}^*,\boldsymbol{\lambda}^*,\boldsymbol{\mu}^*,\boldsymbol{s}^*) = \boldsymbol{h}(\boldsymbol{w}^*) = \mathbf{0} \quad \text{(C.2–8)}$$

$$\nabla_{\boldsymbol{\mu}}\mathcal{L}(\boldsymbol{w}^*,\boldsymbol{\lambda}^*,\boldsymbol{\mu}^*,\boldsymbol{s}^*) = \boldsymbol{g}(\boldsymbol{w}^*) + \mathbf{s}^* \odot \mathbf{s}^* = \mathbf{0} \quad \text{(C.2–9)}$$

$$\nabla_{\boldsymbol{s}}\mathcal{L}(\boldsymbol{w}^*,\boldsymbol{\lambda}^*,\boldsymbol{\mu}^*,\boldsymbol{s}^*) = 2\,\boldsymbol{\mu}^* \odot \mathbf{s}^* = \mathbf{0} \quad \text{(C.2–10)}$$

Multiplying eq (C.2–10) by $\mathbf{s}$ with Hardamard operator and combining the equation with eq (C.2–9), the eq (C.2–10) is equivalently:

$$[\nabla_{\boldsymbol{s}}\mathcal{L}(\boldsymbol{w}^*,\boldsymbol{\lambda}^*,\boldsymbol{\mu}^*,\boldsymbol{s}^*)] \odot \mathbf{s}^* = 2\,\boldsymbol{\mu}^* \odot \mathbf{s}^* \odot \mathbf{s}^* = -2\,\boldsymbol{\mu}^* \odot \boldsymbol{g}(\boldsymbol{w}^*) = \mathbf{0} \quad \text{(C.2–11)}$$

The condition eq (C.2–11) means that, for each inequality constraint $g_j(\boldsymbol{w}^*)$ at $\boldsymbol{w}^*$ with $j = 1$ to q, either $g_j(\boldsymbol{w}^*) = 0$ or $\mu_j = 0$ holds true. If both of them are zero, it should be the trivial case that inequality constraint *j* is not presented in Problem C2.

The inequality constraints with equality sign $g_j(\boldsymbol{w}^*) = 0$ are termed the *active constraints*, denoted as $\mathcal{A}_g(\boldsymbol{w}^*) = \{j \,|g_j(\boldsymbol{w}^*) = 0, j = 1\ to\ q\}$. On the other hand, $\mu_j = 0$ means $g_j(\boldsymbol{w}^*) + s_i^2 = 0$ with $s_i^2 > 0$ and $g_j(\boldsymbol{w}^*) < 0$. The inequality constraints satisfied with strict inequality sign $g_j(\boldsymbol{w}^*) < 0$, and these constraints belong to *inactive constraints* $\mathcal{J}_g(\boldsymbol{w}^*) = \{j \,|g_j(\boldsymbol{w}^*) < 0, j = 1\ to\ q\}$. Following this definition, every equality constraint $h_i(\boldsymbol{w}^*)$ is an *active constraint*, as eq (C.2–8) requires. To differentiate them from inequality *active constraints,* let equality *active constraints* be denoted as $\mathcal{A}_h(\boldsymbol{w}^*) = \{i \,|h_i(\boldsymbol{w}^*) = 0, i = 1\ to\ p\}$. Based on eq (C.2–11), there are q elements in $\mathcal{A}_g(\boldsymbol{w}^*) \cup \mathcal{J}_g(\boldsymbol{w}^*)$. From eqs (C.2–7), (C.2–8), (C.2–9), and (C.2–11), we have $(m + p + q)$ equations to solve for $(m + p + q)$ unknown variables.

doi:10.6342/NTU202403528

Let $\mathbf{A}(\boldsymbol{w}^*)$ be the matrix whose rows are the gradients of the objective function and active constraints at local minimum solution $\boldsymbol{w}^*$, i.e.

$$\boldsymbol{A}(\boldsymbol{w}^*) = \begin{bmatrix} [\nabla_{\mathbf{w}} Objfcn(\boldsymbol{w}^*)]^T{}_{1\times m} \\ [\nabla_{\mathbf{w}} \boldsymbol{h}(\boldsymbol{w}^*)]^T{}_{p\times m}, i \in \mathcal{A}_h(\boldsymbol{w}^*) \\ [\nabla_{\mathbf{w}} \boldsymbol{g}(\boldsymbol{w}^*)]^T{}_{|\mathcal{A}_g(\boldsymbol{w}^*)|\times m}, j \in \mathcal{A}_g(\boldsymbol{w}^*) \end{bmatrix} \tag{C.2–12}$$

Now invoke Gordan's theorem (section C.4). Since $\boldsymbol{w}^*$ is a local minimum solution and $\nabla_{\mathbf{w}} Objfcn(\boldsymbol{w}^*)\boldsymbol{d} \geq 0$ for any feasible direction $\boldsymbol{d}$ (section C.3), the equations $\boldsymbol{A}(\boldsymbol{w}^*)\boldsymbol{d} < \mathbf{0}$ has no solution. Therefore, $\boldsymbol{A^T}(\boldsymbol{w}^*)\mathbf{p} = [\nabla_{\mathbf{w}} Objfcn(\boldsymbol{w}^*), \nabla_{\mathbf{w}} h_i(\boldsymbol{w}^*)\nabla_{\mathbf{w}} g_j(\boldsymbol{w}^*)]\mathbf{p} = \mathbf{0}$ (with $\mathbf{p} \geq \mathbf{0}$) has a solution. Clearly, $\mathbf{p}$ refers to the Lagrange multipliers in column vector form, according to (C.2–7).

$$\mathbf{p} = \begin{bmatrix} 1 \\ [\lambda_i^*]_{p\times 1}, i \in \mathcal{A}_h(\boldsymbol{w}^*) \\ [\mu_j^*]_{|\mathcal{A}_g(\boldsymbol{w}^*)|\times 1}, j \in \mathcal{A}_g(\boldsymbol{w}^*) \end{bmatrix} \geq \mathbf{0} \tag{C.2–13}$$

Nota that, for each inactive inequality constraint, the multipliers are zero, i.e. $\mu_j^* = 0$ for $j \notin \mathcal{A}_g(\boldsymbol{w}^*)$. Also, all the equality constraints are active, i.e. $\mathcal{A}_h(\boldsymbol{w}^*) = \{i \,|i = 1\ to\ p\}$. Consequently, eq (C.2–13) implies that all the multipliers are nonnegative., namely

$$\boldsymbol{\lambda}^* \geq \mathbf{0} \tag{C.2–14}$$

$$\boldsymbol{\mu}^* \geq \mathbf{0} \tag{C.2–15}$$

Finally, the first-order necessary conditions are presented as a system of conditions

doi:10.6342/NTU202403528

consisting of eq (C.2–7) to eq (C.2–9), eq (C.2–11), eq (C.2–14), and eq (C.2–15). If the *active inequality constraints*, $\mathcal{A}_g(\boldsymbol{w}^*)$, are known, these conditions can also be expressed in a more straightforward from:

$$\nabla_{\mathbf{w}}\mathcal{L}(\boldsymbol{w}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*, \boldsymbol{s}^*) = \nabla_{\mathbf{w}} Objfcn(\boldsymbol{w}^*) + \sum_{i=1}^{p} \lambda_i^* \nabla_{\mathbf{w}} h_i(\boldsymbol{w}^*) + \sum_{j=1}^{q} \mu_j^* \nabla_{\mathbf{w}} g_j(\boldsymbol{w}^*) = \mathbf{0} \quad \text{(C.2–16)}$$

$$h_i(\boldsymbol{w}^*) = 0, \qquad \forall i \in \mathcal{A}_h(\boldsymbol{w}^*) \quad \text{(C.2–17)}$$

$$g_j(\boldsymbol{w}^*) = 0, \qquad \forall j \in \mathcal{A}_g(\boldsymbol{w}^*) \quad \text{(C.2–18)}$$

$$\lambda_i^* \geq 0, \qquad \forall i \in \mathcal{A}_h(\boldsymbol{w}^*) \quad \text{(C.2–19)}$$

$$\mu_j^* \geq 0, \qquad \forall j \in \mathcal{A}_g(\boldsymbol{w}^*) \quad \text{(C.2–20)}$$

$$\mu_j^* = 0, \qquad \forall j \in \mathcal{J}_g(\boldsymbol{w}^*) \quad \text{(C.2–21)}$$

with *active constraints* $\mathcal{A}_h(\boldsymbol{w}^*)$ and $\mathcal{A}_g(\boldsymbol{w}^*)$.

$$\mathcal{A}_h(\boldsymbol{w}^*) = \{i \,|h_i(\boldsymbol{w}^*) = 0, i = 1\ to\ p\} \quad \text{(C.2–22)}$$

$$\mathcal{A}_g(\boldsymbol{w}^*) = \left\{j \,\middle|\, g_j(\boldsymbol{w}^*) = 0, j = 1\ to\ q\right\} \quad \text{(C.2–23)}$$

C.2.2. First-order Sufficient Condition (Karush-Kuhn-Tucker, KKT)

Suppose that $\boldsymbol{w}^*$ is a local minimum solution to Problem C2. Following the notation in section C.2.1, let $(\boldsymbol{w}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*, \mathbf{s}^*)$ satisfies the first-order KKT necessary condition with linear independent constraint qualification. Let $\mathcal{A}_g(\boldsymbol{w}^*) = \left\{j \,\middle|\, g_j(\boldsymbol{w}^*) = 0, j = 1 \text{ to } q\right\}$ to indicate the active inequality constraints at $\boldsymbol{w}^*$. Let $\mathcal{A}_h^+(\boldsymbol{w}^*, \boldsymbol{\lambda}^*) = \{i \mid \lambda_i^* > 0, i = 1 \text{ to } p\}$ and $\mathcal{A}_h^-(\boldsymbol{w}^*, \boldsymbol{\lambda}^*) = \{i \mid \lambda_i^* < 0, i = 1 \text{ to } p\}$. If

(I) $Objfcn(\boldsymbol{w}^*)$ is pseudo-convex at $\boldsymbol{w}^*$,

doi:10.6342/NTU202403528

(II) $g_j(\boldsymbol{w}^*)$ with $j \in \mathcal{A}_g(\boldsymbol{w}^*)$ are quasi-convex at $\boldsymbol{w}^*$,

(III) $h_i(\boldsymbol{w}^*)$ with $i \in \mathcal{A}_h^+(\boldsymbol{w}^*)$ are quasi-convex at $\boldsymbol{w}^*$, and

(IV) $h_i(\boldsymbol{w}^*)$ with $i \in \mathcal{A}_h^-(\boldsymbol{w}^*)$ are quasi-concave at $\boldsymbol{w}^*$.

then $\boldsymbol{w}^*$ is a global optimal solution to Problem C2. If these convexity properties are only restricted to a small domain, then $\boldsymbol{w}^*$ is a local minimum.

C.2.3. Second-order Necessary Condition

Suppose that $\boldsymbol{w}^*$ is a local minimum solution to Problem C2. Following the notation in section C.2.1, let $(\boldsymbol{w}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*, \mathbf{s}^*)$ satisfies the first-order KKT necessary condition with linear independent constraint qualification. Then

$$\boldsymbol{d}^T[\nabla_{\boldsymbol{w}}\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{w}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*, \boldsymbol{s}^*)]\boldsymbol{d} \geq 0, \forall \boldsymbol{d} \in \mathcal{F}(\boldsymbol{w}^*) \quad \text{(C.2–24)}$$

C.2.4. Second-order Sufficient Condition

Let $\boldsymbol{w}^*$ be a local minimum solution to Problem C2. Following the notation in section C.2.1, let $(\boldsymbol{w}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*, \mathbf{s}^*)$ satisfies the first-order KKT necessary condition with linear independent constraint qualification. Suppose

$$\boldsymbol{d}^T[\nabla_{\boldsymbol{w}}\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{w}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*, \boldsymbol{s}^*)]\boldsymbol{d} > 0, \forall \boldsymbol{d} \in \mathcal{F}(\boldsymbol{w}^*), \boldsymbol{d} \neq 0 \quad \text{(C.2–25)}$$

Then, $\boldsymbol{w}^*$ is a local minimum solution.

doi:10.6342/NTU202403528

C.3. Tangent Cone and Feasible Directions[393]

Let $\boldsymbol{w_1}$ be a feasible point in a closed convex set S. A sequence $[\mathbf{z_1}, \dots, \mathbf{z_k}]$ with $\mathbf{z_k} \in S$ and $\mathbf{z_k} \to \boldsymbol{w_1}$ for all sufficiently large k is defined as a *feasible sequence approaching* $\boldsymbol{w_1}$. A tangent vector $\boldsymbol{d}$ at $\boldsymbol{w_1}$ is defined as eq (C.3–1), provided that there are a feasible sequence approaching $\boldsymbol{w_1}$ and a corresponding sequence of positive scalars $[t_1, \dots, t_k]$ with $t_k \to 0$.

$$\lim_{k\to\infty} \frac{\boldsymbol{z_k} - \boldsymbol{w_1}}{t_k} = \boldsymbol{d} \tag{C.3–1}$$

Let the active constraint set at $\boldsymbol{w_1}$ be $\mathcal{A}(\boldsymbol{w_1}) = \mathcal{A}_h(\boldsymbol{w_1}) \cup \mathcal{A}_g(\boldsymbol{w_1})$ (see section C.2.1 for the definition of $\mathcal{A}_g(\boldsymbol{w_1})$ and $\mathcal{A}_h(\boldsymbol{w_1})$). The set of feasible directions $\mathcal{F}(\boldsymbol{w_1})$ for Problem C2, is defined as:

$$\mathcal{F}(\boldsymbol{w_1}) = \left\{ \boldsymbol{d} \;\middle|\; \begin{matrix} \boldsymbol{d^T \nabla} h_i(\boldsymbol{w_1}) = \boldsymbol{0} \;\; \forall i \in \mathcal{A}_h(\boldsymbol{w_1}) \;\; and \\ \boldsymbol{d^T \nabla} g_j(\boldsymbol{w_1}) \leq \boldsymbol{0} \;\; \forall j \in \mathcal{A}_g(\boldsymbol{w_1}) \end{matrix} \right\} \tag{C.3–2}$$

The central ideal of this definition is to collect possible directions $\boldsymbol{d}$ along which ensures an optimization step will not move $\boldsymbol{w_1}$ out of feasible region. To see this, one can utilize the definition of tangent vector (eq (C.3–1)) and regard $\boldsymbol{z_k}$ as the point after an optimization step is conducted to $\boldsymbol{w_1}$.

$$h_i(\boldsymbol{z_k}) = h_i(\boldsymbol{w_1}) + t_k \boldsymbol{d^T \nabla_w} h_i(\boldsymbol{w_1}) + \mathcal{O}(t_k) \tag{C.3–3}$$

$$g_j(\boldsymbol{z_k}) = g_j(\boldsymbol{w_1}) + t_k \boldsymbol{d^T \nabla_w} g_j(\boldsymbol{w_1}) + \mathcal{O}(t_k) \tag{C.3–4}$$

doi:10.6342/NTU202403528

Here, $\mathcal{O}(t_k)$ means that the order of the remaining terms is roughly t_k, with $\lim_{k\to\infty} \mathcal{O}(t_k) = 0$. Based on the constraints in Problem C2, we should require $h_i(\boldsymbol{z_k}) = h_i(\boldsymbol{w_1})$ and $g_j(\boldsymbol{z_k}) \le g_j(\boldsymbol{w_1})$. Therefore, we have from (C.3–3) and (C.3–4) that:

$$\lim_{k\to\infty} \frac{h_i(\boldsymbol{z_k}) - h_i(\boldsymbol{w_1})}{t_k} = \boldsymbol{d}^T \nabla_{\mathbf{w}} h_i(\boldsymbol{w_1}) = 0 \tag{C.3–5}$$

$$\lim_{k\to\infty} \frac{g_j(\boldsymbol{z_k}) - g_j(\boldsymbol{w_1})}{t_k} = \boldsymbol{d}^T \nabla_{\mathbf{w}} g_j(\boldsymbol{w_1}) \le 0 \tag{C.3–6}$$

The same treatment can be applied to the objective function, and we will have:

$$Objfcn(\boldsymbol{z_k}) = Objfcn(\boldsymbol{w_1}) + t_k \boldsymbol{d}^T \nabla_{\mathbf{w}} Objfcn(\boldsymbol{w_1}) + \mathcal{O}(t_k) \tag{C.3–7}$$

It should be noted that, from eq (C.3–2), the optimization of $\boldsymbol{w_1}$ along a feasible direction $\boldsymbol{d}$ does not guarantee further improvement of $Objfcn(\boldsymbol{w_1})$, as such property is not imposed in the definition. If $\boldsymbol{w_1}$ is away from any local minimum solution, it is desirable to require $Objfcn(\boldsymbol{z_k}) \le Objfcn(\boldsymbol{w_1})$, or equivalently $\boldsymbol{d}^T \nabla_{\mathbf{w}} Objfcn(\boldsymbol{w_1}) \le \mathbf{0}$.

$$\lim_{k\to\infty} \frac{Objfcn(\boldsymbol{z_k}) - Objfcn(\boldsymbol{w_1})}{t_k} = \boldsymbol{d}^T \nabla_{\mathbf{w}} Objfcn(\boldsymbol{w_1}) \le 0 \tag{C.3–8}$$

From this, the set of usable feasible directions $\mathcal{F}_u(\boldsymbol{w_1})$, also called the set of improving feasible directions, is defined as:

doi:10.6342/NTU202403528

$$\mathcal{F}_u(\boldsymbol{w_1}) = \mathcal{F}(\boldsymbol{w_1}) \cap \{\boldsymbol{d} \mid \boldsymbol{d^T \nabla_w} Objfcn(\boldsymbol{w_1}) \leq \boldsymbol{0}\} \quad \text{(C.3–9)}$$

Every improving feasible directions should be capable of minimizing $Objfcn(\boldsymbol{w_1})$ if $\boldsymbol{w_1}$ is not a local minimum solution. If $\boldsymbol{w}^*$ is a local minimum solution to the minimization problem, then for every $\boldsymbol{d} \in \mathcal{F}(\boldsymbol{w}^*)$ we have $\boldsymbol{d^T \nabla_w} Objfcn(\boldsymbol{w}^*) \geq \boldsymbol{0}$. This can be seen from eq (C.3–7).

C.4. Gordan's Theorem[394]

Let $\mathbf{A}$ be an $m \times n$ matrix. Then either of the following systems has a solution.

System 1: $\mathbf{Ax} < \mathbf{0}$ for some $\mathbf{x} \in \mathbb{R}^n$

System 2: $\mathbf{A^T p} = \mathbf{0}$ *with* $\mathbf{p} \geq \mathbf{0}$ for some nonzero $\mathbf{p} \in \mathbb{R}^m$

C.5. Lagrangian Duality Problem[394]

Every nonlinear primal problem C5 has a Lagrangian dual Problem Dual(C5)

Problem C5

$$\min_{\boldsymbol{w}} Objfcn(\boldsymbol{w}) \quad \text{(C.5–1)}$$

subjected to

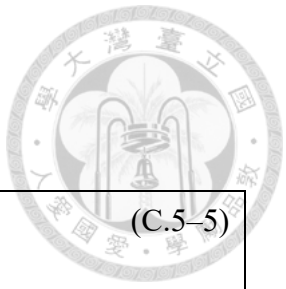
$$\boldsymbol{h}(\boldsymbol{w}) = 0 \quad \text{(C.5–2)}$$

$$\boldsymbol{g}(\boldsymbol{w}) \leq 0 \quad \text{(C.5–3)}$$

$$\boldsymbol{w} \in W \subseteq \mathbb{R}^m \quad \text{(C.5–4)}$$

Denote the Lagrange function for Problem C5 as $\theta(\boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = Objfcn(\boldsymbol{w}) + \sum_{i=1}^{p} \lambda_i h_i(\boldsymbol{w}) + \sum_{j=1}^{q} \mu_j g_j(\boldsymbol{w})$. (Note: slack variables are excluded)

doi:10.6342/NTU202403528



Problem Dual(C5)

$$\max_{\boldsymbol{\lambda},\boldsymbol{\mu}\geq\mathbf{0}}[\inf_{\boldsymbol{w}}\theta(\boldsymbol{w},\boldsymbol{\lambda},\boldsymbol{\mu})] \tag{C.5–5}$$

subjected to

$$\boldsymbol{\mu}\geq\mathbf{0} \tag{C.5–6}$$

$$\boldsymbol{\lambda}\in\mathbb{R}^{p} \tag{C.5–7}$$

$$\boldsymbol{\mu}\in\mathbb{R}^{q} \tag{C.5–8}$$

C.6. Nonlinear Duality Theorem[394]

Let $\boldsymbol{w}$ be a feasible solution to Problem C5, i.e. $\boldsymbol{w}\in W\subseteq\mathbb{R}^{m}$, $\boldsymbol{g}(\boldsymbol{w})\leq 0$, and $\boldsymbol{h}(\boldsymbol{w})=0$. Also, let $(\boldsymbol{\lambda},\boldsymbol{\mu})$ be a feasible solution to Problem Dual(C5), i.e. $\boldsymbol{\mu}\geq\mathbf{0}$. Then

$$Objfcn(\boldsymbol{w})\geq\sup_{\boldsymbol{\lambda},\boldsymbol{\mu}\geq\mathbf{0}}[\inf_{\boldsymbol{w}}\theta(\boldsymbol{w},\boldsymbol{\lambda},\boldsymbol{\mu})]\geq\inf_{\boldsymbol{w}}\theta(\boldsymbol{w},\boldsymbol{\lambda},\boldsymbol{\mu}) \tag{C.6–1}$$

Since $\sum_{i=1}^{p}\lambda_i h_i(\boldsymbol{w})+\sum_{j=1}^{q}\mu_j g_j(\boldsymbol{w})\leq 0$ for all the feasible solution $\boldsymbol{w}$, we have $\inf_{\boldsymbol{w}}\theta(\boldsymbol{w},\boldsymbol{\lambda},\boldsymbol{\mu})\leq\inf_{\boldsymbol{w}}Objfcn(\boldsymbol{w})\leq Objfcn(\boldsymbol{w})$. In particular, $\inf_{\boldsymbol{w}}Objfcn(\boldsymbol{w})=Objfcn(\boldsymbol{w}^{*})$, where $\boldsymbol{w}^{*}$ is an optimal solution to Problem C5. Also note that $\sum_{i=1}^{p}\lambda_i^{*}h_i(\boldsymbol{w}^{*})+\sum_{j=1}^{q}\mu_j^{*}g_j(\boldsymbol{w}^{*})=0$ since KKT conditions are satisfied at $\boldsymbol{w}^{*}$**.** From this, the minimum of Problem C5 at $\boldsymbol{w}^{*}$ will be the same as the maximum of Problem Dual(C5) at $\boldsymbol{w}^{*}$.

$$\min_{\boldsymbol{w}}Objfcn(\boldsymbol{w})=Objfcn(\boldsymbol{w}^{*})=\sup_{\boldsymbol{\lambda},\boldsymbol{\mu}\geq\mathbf{0}}[\inf_{\boldsymbol{w}}\theta(\boldsymbol{w},\boldsymbol{\lambda},\boldsymbol{\mu})]=\theta(\boldsymbol{w}^{*},\boldsymbol{\lambda}^{*},\boldsymbol{\mu}^{*}) \tag{C.6–2}$$

doi:10.6342/NTU202403528

Appendix D. Solving Non-linear Programming Problems[68]

Problem D

$$\underset{\boldsymbol{w}}{\operatorname{argmin}}\ Objfcn(\boldsymbol{w}) \tag{C.6–1}$$

subjected to

$$\boldsymbol{h}(\boldsymbol{w}) = 0 \tag{C.6–2}$$

$$\boldsymbol{g}(\boldsymbol{w}) \leq 0 \tag{C.6–3}$$

$$\boldsymbol{w} \in W \subseteq \mathbb{R}^m \tag{C.6–4}$$

D.1. Sequential Quadratic Programming (SQP) Method

Recall the first-order necessary KKT conditions, eqs (D.1–1) to (D.1–4):

$$\nabla_{\mathbf{w}}\mathcal{L}(\boldsymbol{w}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*, \boldsymbol{s}^*) = \nabla_{\mathbf{w}} Objfcn(\boldsymbol{w}^*) + \sum_{i=1}^{p} \lambda_i^* \nabla_{\mathbf{w}} h_i(\boldsymbol{w}^*) + \sum_{j=1}^{q} \mu_j^* \nabla_{\mathbf{w}} g_j(\boldsymbol{w}^*) = \mathbf{0} \tag{D.1–1}$$

$$\nabla_{\boldsymbol{\lambda}}\mathcal{L}(\boldsymbol{w}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*, \boldsymbol{s}^*) = \boldsymbol{h}(\boldsymbol{w}^*) = \mathbf{0} \tag{D.1–2}$$

$$\nabla_{\boldsymbol{\mu}}\mathcal{L}(\boldsymbol{w}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*, \boldsymbol{s}^*) = \boldsymbol{g}(\boldsymbol{w}^*) + \mathbf{s}^* \odot \mathbf{s}^* = \mathbf{0} \tag{D.1–3}$$

$$\frac{1}{2}\nabla_{\boldsymbol{s}}\mathcal{L}(\boldsymbol{w}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*, \boldsymbol{s}^*) = \boldsymbol{\mu}^* \odot \mathbf{s}^* = \mathbf{0} \tag{D.1–4}$$

Sequential Quadratic Programming (SQP) employs the Newton method (or quasi-Newton method, depending on the Hessian update scheme) to seek a solution $(\boldsymbol{w}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*, \boldsymbol{s}^*)$ satisfying the first-order necessary KKT conditions. To simplify the formulation, let us introduce the variable $\boldsymbol{Y}$ and function $\boldsymbol{F}(\boldsymbol{Y})$ as eqs (D.1–5) and (D.1–6), respectively.

doi:10.6342/NTU202403528

$$\boldsymbol{Y} = \begin{bmatrix} [\boldsymbol{w}]_{m\times 1} \\ [\boldsymbol{\lambda}]_{p\times 1} \\ [\boldsymbol{\mu}]_{q\times 1} \\ [\boldsymbol{s}]_{q\times 1} \end{bmatrix} \tag{D.1–5}$$

$$\boldsymbol{F}(\boldsymbol{Y}) = \begin{bmatrix} [\nabla_{\mathbf{w}}\mathcal{L}(\boldsymbol{Y})]_{m\times 1} \\ [\boldsymbol{h}(\boldsymbol{Y})]_{p\times 1} \\ [\boldsymbol{g}(\boldsymbol{Y}) + \mathbf{s} \odot \mathbf{s}]_{q\times 1} \\ [\boldsymbol{\mu} \odot \mathbf{s}]_{q\times 1} \end{bmatrix} \tag{D.1–6}$$

From eqs (D.1–1) to (D.1–4), $\boldsymbol{F}(\boldsymbol{Y}^*) = \mathbf{0}$ at an optimal solution $\boldsymbol{Y}^*$. The Newton iteration scheme is formulated as follows:

$$[\boldsymbol{\nabla_Y F}(\boldsymbol{Y_k})]\boldsymbol{\Delta Y_k} = [\boldsymbol{\nabla_Y F}(\boldsymbol{Y_k})](\boldsymbol{Y_{k+1}} - \boldsymbol{Y_k}) = -\boldsymbol{F}(\boldsymbol{Y_k}) \tag{D.1–7}$$

Here, $\boldsymbol{\Delta Y_k} = (\boldsymbol{Y_{k+1}} - \boldsymbol{Y_k})$. $\boldsymbol{Y_{k+1}}$ is the feasible solution obtained from at the *(k+1)-th* iteration, determined based on the function value $\boldsymbol{F}(\boldsymbol{Y_k})$ and gradients $\boldsymbol{\nabla_Y F}(\boldsymbol{Y_k})$ at the previous solution $\boldsymbol{Y_k}$. In particular, the gradients $\boldsymbol{\nabla_Y F}(\boldsymbol{Y_k})$ take the form:

$$[\boldsymbol{\nabla_Y F}(\boldsymbol{Y_k})] = \begin{bmatrix} \boldsymbol{B}_{(m+p+q)\times(m+p+q)} & \boldsymbol{C}_{(m+p+q)\times q} \\ \boldsymbol{D}_{q\times(m+p+q)} & \boldsymbol{E}_{q\times q} \end{bmatrix} \tag{D.1–8}$$

The matrices $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{D}$, and $\boldsymbol{E}$ are:

$$\boldsymbol{B} = \begin{bmatrix} [\nabla_{\mathbf{w}}\nabla_{\mathbf{w}}\mathcal{L}(\boldsymbol{Y_k})]_{m\times m} & [\nabla_{\mathbf{w}}\boldsymbol{h}(\boldsymbol{Y_k})]_{m\times p} & [\nabla_{\mathbf{w}}\boldsymbol{g}(\boldsymbol{Y_k})]_{m\times q} \\ [\nabla_{\mathbf{w}}\boldsymbol{h}(\boldsymbol{Y_k})]^T{}_{p\times m} & \mathbf{0} & \mathbf{0} \\ [\nabla_{\mathbf{w}}\boldsymbol{g}(\boldsymbol{Y_k})]^T{}_{q\times m} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{D.1–9}$$

doi:10.6342/NTU202403528

$$\boldsymbol{C} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ 2\mathbf{s} \end{bmatrix} \tag{D.1–10}$$

$$\boldsymbol{D} = [\mathbf{0} \quad \mathbf{0} \quad \boldsymbol{s}^{\boldsymbol{T}}\boldsymbol{I}] \tag{D.1–11}$$

$$\boldsymbol{E} = [\boldsymbol{\mu}^{\boldsymbol{T}}\boldsymbol{I}] \tag{D.1–12}$$

Here, $\boldsymbol{I}$ denotes the identity matrix. Iterations proceed from $\boldsymbol{Y_k} \rightarrow \boldsymbol{Y_{k+1}}$ via eq (D.1–7) continue until $\boldsymbol{F}(\boldsymbol{Y_{k+1}}) = \mathbf{0}$ is satisfied. Alternatively, Problem D can be reformulated into a sequence of a sequence of quadratic programming (QP) Problem D1 by linearizing of constraints.[395]

Problem D1 (Quadratic programming with linearized constraints)

$$\underset{\boldsymbol{\Delta w_k}}{\operatorname{argmin}}\ Objfcn(\boldsymbol{w_k}) + [\nabla_{\mathbf{w}} Objfcn(\boldsymbol{w_k})]\boldsymbol{d} + \frac{1}{2}\boldsymbol{d}^{\boldsymbol{T}}[\nabla_{\mathbf{w}}\nabla_{\mathbf{w}}\mathcal{L}(\boldsymbol{w_k})]\boldsymbol{\Delta w_k} \tag{D.1–13}$$

subjected to

$$\boldsymbol{h}(\boldsymbol{w_k}) + [\nabla_{\mathbf{w}}\boldsymbol{h}(\boldsymbol{w_k})]\boldsymbol{\Delta w_k} = 0 \tag{D.1–14}$$

$$\boldsymbol{g}(\boldsymbol{w_k}) + [\nabla_{\mathbf{w}}\boldsymbol{g}(\boldsymbol{w_k})]\boldsymbol{\Delta w_k} \leq 0 \tag{D.1–15}$$

$$\boldsymbol{\Delta w_k} \in W \subseteq \mathbb{R}^m \tag{D.1–16}$$

From Lagrange multiplier formulation (eq (D.1–17)), one of the first-order necessary KKT conditions for Problem D1 is eq (D.1–18), and it is equivalent to the first *m* rows of $[\boldsymbol{\nabla_Y F}(\boldsymbol{Y_k})]\boldsymbol{\Delta Y_k}$. This condition justifies that Problem D can be solved iteratively by solving Problem D1 at each newly traversed $\boldsymbol{w_k}$.

doi:10.6342/NTU202403528

$$\tilde{\mathcal{L}}(\boldsymbol{\Delta w_k}, \tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\mu}}) = Objfcn(\boldsymbol{w_k}) + [\nabla_{\mathbf{w}} Objfcn(\boldsymbol{w_k})]\boldsymbol{d} + \frac{1}{2}\boldsymbol{d^T}[\nabla_{\mathbf{w}}\nabla_{\mathbf{w}}\mathcal{L}(\boldsymbol{w_k})]\boldsymbol{\Delta w_k}$$
$$+ \sum_{i=1}^{p} \tilde{\lambda}_i^* [h_i(\boldsymbol{w_k}) + [\nabla_{\mathbf{w}} h_i(\boldsymbol{w_k})]\boldsymbol{\Delta w_k}] \qquad \text{(D.1–17)}$$
$$+ \sum_{j=1}^{q} \widetilde{\mu_j^*}\left[g_j(\boldsymbol{w_k}) + \left[\nabla_{\mathbf{w}} g_j(\boldsymbol{w_k})\right]\boldsymbol{\Delta w_k}\right]$$

$$\boldsymbol{\nabla_d \tilde{\mathcal{L}}}(\boldsymbol{\Delta w_k}, \widetilde{\boldsymbol{\lambda^*}}, \widetilde{\boldsymbol{\mu^*}})$$

$$= \nabla_{\mathbf{w}} Objfcn(\boldsymbol{w_k}) + [\nabla_{\mathbf{w}}\nabla_{\mathbf{w}}\mathcal{L}(\boldsymbol{w_k})]\boldsymbol{\Delta w_k} + \sum_{i=1}^{p} \tilde{\lambda}_i^* \nabla_{\mathbf{w}} h_i(\boldsymbol{w_k}) + \sum_{j=1}^{q} \widetilde{\mu_j^*} \nabla_{\mathbf{w}} g_j(\boldsymbol{w_k}) \qquad \text{(D.1–18)}$$

$$= \mathbf{0}$$

doi:10.6342/NTU202403528

Appendix E. Generalized Benders Decomposition[37, 69, 70]

Problem E

$\underset{\boldsymbol{u},\boldsymbol{w}}{\operatorname{argmin}} \, Objfcn(\boldsymbol{u},\boldsymbol{w})$	(D.1–1)
subjected to	
$\boldsymbol{h}(\boldsymbol{u},\boldsymbol{w}) = 0$	(D.1–2)
$\boldsymbol{g}(\boldsymbol{u},\boldsymbol{w}) \leq 0$	(D.1–3)
$\boldsymbol{u} \in U \subseteq \mathbb{Z}^n$	(D.1–4)
$\boldsymbol{w} \in W \subseteq \mathbb{R}^m$	(D.1–5)

E.1. Problem Projection

Problem E can be transformed into a $\boldsymbol{u}$-space Problem $E_{\boldsymbol{u}}$:

Problem $E_{\boldsymbol{u}}$

$\underset{\boldsymbol{u}}{\operatorname{argmin}} \, \nu(\boldsymbol{u})$	(E.1–1)
subjected to	
$\boldsymbol{u} \in V$	(E.1–2)
with $V = \{\boldsymbol{u} \mid \boldsymbol{h}(\boldsymbol{u},\boldsymbol{w}) = 0, \boldsymbol{g}(\boldsymbol{u},\boldsymbol{w}) \leq 0 \text{ for some } \boldsymbol{w} \in W\}$	

Here, $\nu(\boldsymbol{u})$ is named Problem Dual(E_u), signifying that it is the dual problem (see section C.5) of Problem $E_{\boldsymbol{u}}$. Problem Dual(E_u) is a nonlinear programming problem where $\boldsymbol{w}$ is the variable to be optimized and $\boldsymbol{u}$ is fixed at some value. This dual problem is presented in the next section.

doi:10.6342/NTU202403528

E.2. Dual Problems of the Projected Problems

Problem Dual(E_u)

$$\nu(\boldsymbol{u}) = \inf_{\boldsymbol{w}} Objfcn(\boldsymbol{u}, \boldsymbol{w}) \quad \text{(E.2–1)}$$

subjected to

$$\boldsymbol{h}(\boldsymbol{u}, \boldsymbol{w}) = 0 \quad \text{(E.2–2)}$$

$$\boldsymbol{g}(\boldsymbol{u}, \boldsymbol{w}) \leq 0 \quad \text{(E.2–3)}$$

$$\boldsymbol{w} \in W \subseteq \mathbb{R}^m \quad \text{(E.2–4)}$$

Formulate the dual problem of Problem Dual(E_u), denoted as Problem Dual^2(E_u), with $\boldsymbol{u}$ still fixed at the same value. Let $\theta(\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = Objfcn(\boldsymbol{u}, \boldsymbol{w}) + \sum_{i=1}^{p} \lambda_i h_i(\boldsymbol{u}, \boldsymbol{w}) + \sum_{j=1}^{q} \mu_j g_j(\boldsymbol{u}, \boldsymbol{w})$.

Problem Dual^2(E_u)

$$\sup_{\boldsymbol{\lambda}, \boldsymbol{\mu} \geq \boldsymbol{0}} [\inf_{\boldsymbol{w}} \theta(\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{\mu})] \quad \text{(E.2–5)}$$

subjected to

$$\boldsymbol{\mu} \geq \boldsymbol{0} \quad \text{(E.2–6)}$$

$$\boldsymbol{u} \in V \quad \text{(E.2–7)}$$

$$\boldsymbol{\lambda} \in \mathbb{R}^p \quad \text{(E.2–8)}$$

$$\boldsymbol{\mu} \in \mathbb{R}^q \quad \text{(E.2–9)}$$

E.3. Formulation of GBD Form

Substituting Problem Dual^2(E_u) back to Problem $E_{\boldsymbol{u}}$, we have Problem GBD(E):

Problem GBD(E)

doi:10.6342/NTU202403528

$$\underset{\boldsymbol{u}}{\operatorname{argmin}} \left[\sup_{\boldsymbol{\lambda},\boldsymbol{\mu}\geq \mathbf{0}} \left[\inf_{\boldsymbol{w}} \theta(\boldsymbol{u},\boldsymbol{w},\boldsymbol{\lambda},\boldsymbol{\mu}) \right] \right] \tag{E.3–1}$$

subjected to

$$\boldsymbol{u} \in V \tag{E.3–2}$$

$$\boldsymbol{\mu} \geq \mathbf{0} \tag{E.3–3}$$

$$\boldsymbol{\lambda} \in \mathbb{R}^p \tag{E.3–4}$$

$$\boldsymbol{\mu} \in \mathbb{R}^q \tag{E.3–5}$$

with $V = \{\boldsymbol{u} \,|\boldsymbol{h}(\boldsymbol{u},\boldsymbol{w}) = 0, \boldsymbol{g}(\boldsymbol{u},\boldsymbol{w}) \leq 0 \text{ for some } \boldsymbol{w} \in W\}$

Let $\alpha = \sup_{\boldsymbol{\lambda},\boldsymbol{\mu}\geq \mathbf{0}} \left[\inf_{\boldsymbol{w}} \theta(\boldsymbol{u},\boldsymbol{w},\boldsymbol{\lambda},\boldsymbol{\mu}) \right] \leq \inf_{\boldsymbol{w}} Objfcn(\boldsymbol{u},\boldsymbol{w})$ be a lower bound of Problem E under the condition that $\boldsymbol{u}$ is fixed. Reformulate Problem GBD(E):

Problem GBD(E)-M-INLP($\boldsymbol{w_t}$) (GBD master problem, integer nonlinear programming)

$$\underset{\boldsymbol{u},\alpha}{\operatorname{argmin}}\ \alpha \tag{E.3–6}$$

subjected to

$$\alpha \geq \inf_{\boldsymbol{w}} \theta(\boldsymbol{u},\boldsymbol{w},\boldsymbol{\lambda},\boldsymbol{\mu}) \tag{E.3–7}$$

$$\boldsymbol{u} \in V \tag{E.3–8}$$

$$\boldsymbol{\mu} \geq \mathbf{0} \tag{E.3–9}$$

$$\boldsymbol{\lambda} \in \mathbb{R}^p \tag{E.3–10}$$

$$\boldsymbol{\mu} \in \mathbb{R}^q \tag{E.3–11}$$

with $V = \{\boldsymbol{u} \,|\boldsymbol{h}(\boldsymbol{u},\boldsymbol{w}) = 0, \boldsymbol{g}(\boldsymbol{u},\boldsymbol{w}) \leq 0 \text{ for some } \boldsymbol{w} \in W\}$

Note that the $\inf_{\boldsymbol{w}} \theta(\boldsymbol{u},\boldsymbol{w},\boldsymbol{\lambda},\boldsymbol{\mu})$ in eq (E.3–7) represents Problem Dual(E_u), since

doi:10.6342/NTU202403528

$\inf_{\boldsymbol{w}} \theta(\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ characterizes the lower bound of $\inf_{\boldsymbol{w}} Objfcn(\boldsymbol{u}, \boldsymbol{w})$ in Problem Dual(E_u). This problem is also called the Problem GBD(E)-P-NLP($\boldsymbol{u}$).

Problem GBD(E)-P-NLP($\boldsymbol{u}$) (GBD primal problem, nonlinear programming)

$$v(\boldsymbol{u}) = \inf_{\boldsymbol{w}} Objfcn(\boldsymbol{u}, \boldsymbol{w}) \quad \text{(E.3–12)}$$

subjected to

$$\boldsymbol{h}(\boldsymbol{u}, \boldsymbol{w}) = 0 \quad \text{(E.3–13)}$$

$$\boldsymbol{g}(\boldsymbol{u}, \boldsymbol{w}) \leq 0 \quad \text{(E.3–14)}$$

$$\boldsymbol{w} \in W \subseteq \mathbb{R}^m \quad \text{(E.3–15)}$$

E.4. GBD Algorithm

Step 1: Guess initial point $\boldsymbol{u_1}$, solve nonlinear programing problem Problem GBD(E)-P-NLP($\boldsymbol{u}$) (i.e. Problem Dual(E_u), $v(\mathbf{u_1})$). Obtain an optimal (or near-optimal) primal solution $\boldsymbol{w_1}$ as well as the multiplier vectors ($\boldsymbol{\lambda_1}, \boldsymbol{\mu_1}$). Set the counter $k = 1$ if feasible, $r = 1$ if infeasible, and current upper bound $Z_U = v(\boldsymbol{u_1}) = \inf_{\boldsymbol{w}} Objfcn(\boldsymbol{u_1}, \boldsymbol{w})$ of Problem GBD(E)-P-NLP($\boldsymbol{u}$). Set the convergence tolerance $\epsilon \geq 0$.

Step 2: Solve Problem GBD(E)-M-INLP($\boldsymbol{w_t}$), with $\boldsymbol{w}$ fixed at each $\boldsymbol{w_t}$, $t = 1, \ldots, k$. In particular, α should be large than the value of Lagrangian for all the points of ($\boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{\mu}$) that have been tried.

$$\alpha \geq \theta(\boldsymbol{u}, \boldsymbol{w_t}, \boldsymbol{\lambda_t}, \boldsymbol{\mu_t}), \text{for every } t = 1, \ldots, k \quad \text{(E.4–1)}$$

In addition, all the $\boldsymbol{w}$'s that have been tried need to fulfill the requirements of V.

doi:10.6342/NTU202403528

$\forall \boldsymbol{w_t}$ with $t = 1, \dots, r, \quad \exists \boldsymbol{u} \in V$ (E.4–2)

Let $(\boldsymbol{u}^*, \alpha^*)$ be the optimal solution. $Z_L = \alpha^* = \sup_{\boldsymbol{\lambda_t}, \boldsymbol{\mu_t} \geq \mathbf{0}} \left[\inf_{\boldsymbol{w_t}} \theta(\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \right] \leq \inf_{\boldsymbol{w}} Objfcn(\boldsymbol{u}, \boldsymbol{w})$ is a lower bound of the Problem E. If $Z_L + \epsilon \geq Z_U$, terminate iterations.

Step 3: Solve nonlinear programing problem Problem GBD(E)-P-NLP($\boldsymbol{u}$) with $\boldsymbol{u}$ fixed at $\boldsymbol{u}^*$ (i.e. Problem Dual(E_u), $v(\boldsymbol{u}^*)$). One of the following scenarios must occur:

(I) $Z_U^* = v(\boldsymbol{u}^*) = \inf_{\boldsymbol{w}} Objfcn(\boldsymbol{u}^*, \boldsymbol{w})$ is finite with an optimal solution $(\boldsymbol{w}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$. If $Z_L + \epsilon \geq Z_U^*$, terminate the iterations. Otherwise, set $k = k + 1$, $\boldsymbol{w_k} = \boldsymbol{w}^*$ and $\boldsymbol{u_k} = \boldsymbol{u}^*$. If $Z_U^* < Z_U$, set $Z_U = Z_U^*$. Return to Step 2.

(II) Problem $v(\boldsymbol{u}^*) = \inf_{\boldsymbol{w}} Objfcn(\boldsymbol{u}^*, \boldsymbol{w})$ is infeasible for $\boldsymbol{u} = \boldsymbol{u}^*$. This means that $(\boldsymbol{w}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ fails to fulfill the requirements of V at $\boldsymbol{u}^*$. Add $(\boldsymbol{w}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ into the consideration in V, so in next iterations $\boldsymbol{u}$ is determined based $\boldsymbol{w}^*$ on such that $\boldsymbol{u} \in V$. Set $r = r + 1$, $\boldsymbol{w_r} = \boldsymbol{w}^*$, $\boldsymbol{\lambda_r} = \boldsymbol{\lambda}^*$, and $\boldsymbol{\mu_r} = \boldsymbol{\mu}^*$. Return to Step 2.

doi:10.6342/NTU202403528

Appendix F. Theories of Some AI-based Generative Models

F.1. Neuron and Neural Network (NN)

A neural network (NN) consists of multiple layers of neurons, as illustrated by **Figure F1**. Each of the neuron can receive input data $\boldsymbol{x}$ and transform them into an output y through mathematical operations. Specifically, each input data x_i in $\boldsymbol{x}$ is associated with a weight W_i in the neuron. Based on these weights and a bias b, all the input data are lumped into a linear combination form X, as presented in eq. (F.1–1). Then, the X is encoded by a non-linear activation function $y = \sigma(X)$ such as sigmoid function, as presented in eq. (F.1–2). The purpose of activation function is to normalize the X such that the output y is within desirable upper and lower bounds. Finally, the output y serves as the input for the next-layer neurons. It should be noted that the non-linearity of activation function plays a significant role in strengthening the applicability of NNs. With that non-linearity, the NN will be a universal approximator for functions, as stated in universal approximation theorem.[396]

$$X = \boldsymbol{W}\boldsymbol{x} + b = b + \sum_{i=1}^{n} W_i x_i \tag{F.1–1}$$

$$y = \sigma(X) = \frac{1}{1 + \exp(-X)} \tag{F.1–2}$$

In the training process of a NN model, the optimizer iteratively adjusts the weights and bias of each neuron such that the network can reproduce the input-output relation of the training data. This is typically achieved by gradient-based minimization of loss function $\mathcal{L}(\boldsymbol{y}(\boldsymbol{x}); \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ denotes all the adjustable weights and biases in the NN.

doi:10.6342/NTU202403528

There are many gradient-based algorithms for updating the parameters. For example, the newton's method is one of the simplest algorithms based on the first-order and second-order derivatives (i.e. Hessian matrix $\boldsymbol{H}$) of loss function.[397]

$$\boldsymbol{\Delta\theta} = -\boldsymbol{H}^{-1}\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{y}(\boldsymbol{x}); \boldsymbol{\theta}) \tag{F.1–3}$$

Since the gradients with respect to the parameters of layer j (i.e. $\boldsymbol{\nabla}_{\boldsymbol{\theta}_j}\mathcal{L}$) depend on the parameters of all its succeeding layers (i.e. $\boldsymbol{\theta}_{\boldsymbol{k>j}}$), it is much efficient to calculate gradients starting with the output layer and ending up with the input layer. This is known as the back-propagation algorithm.[398, 399]

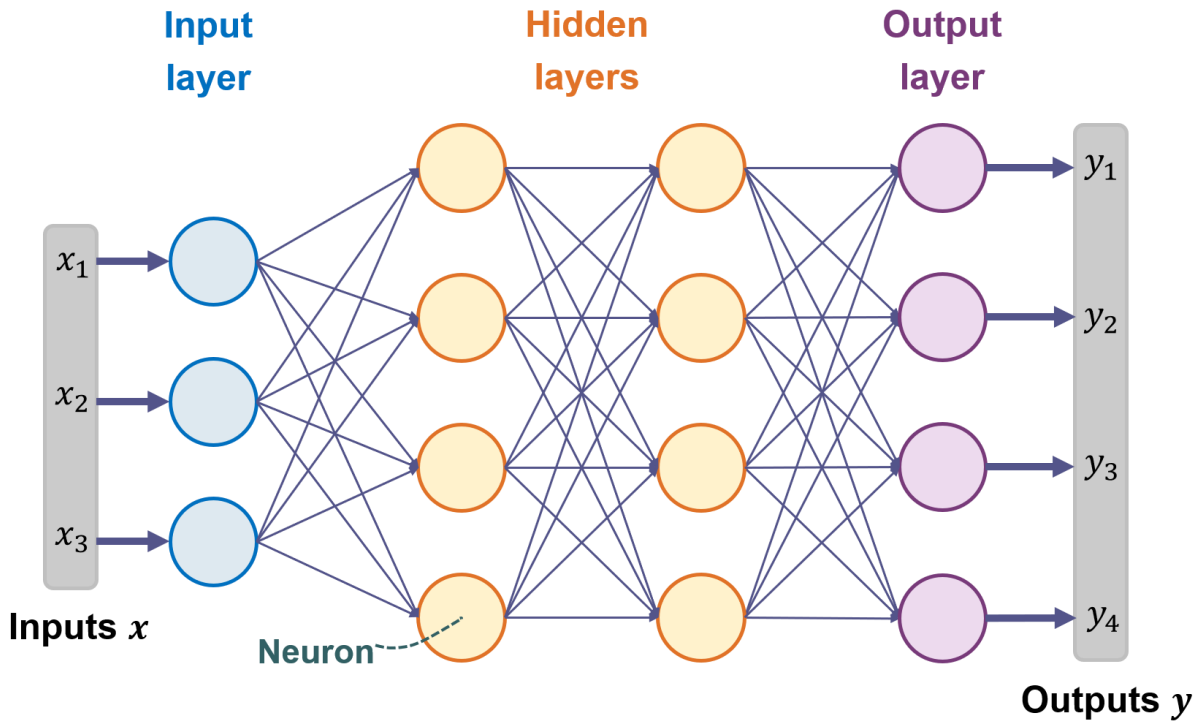


Figure F1. The schematic diagram of a fully-connected neural network.

doi:10.6342/NTU202403528

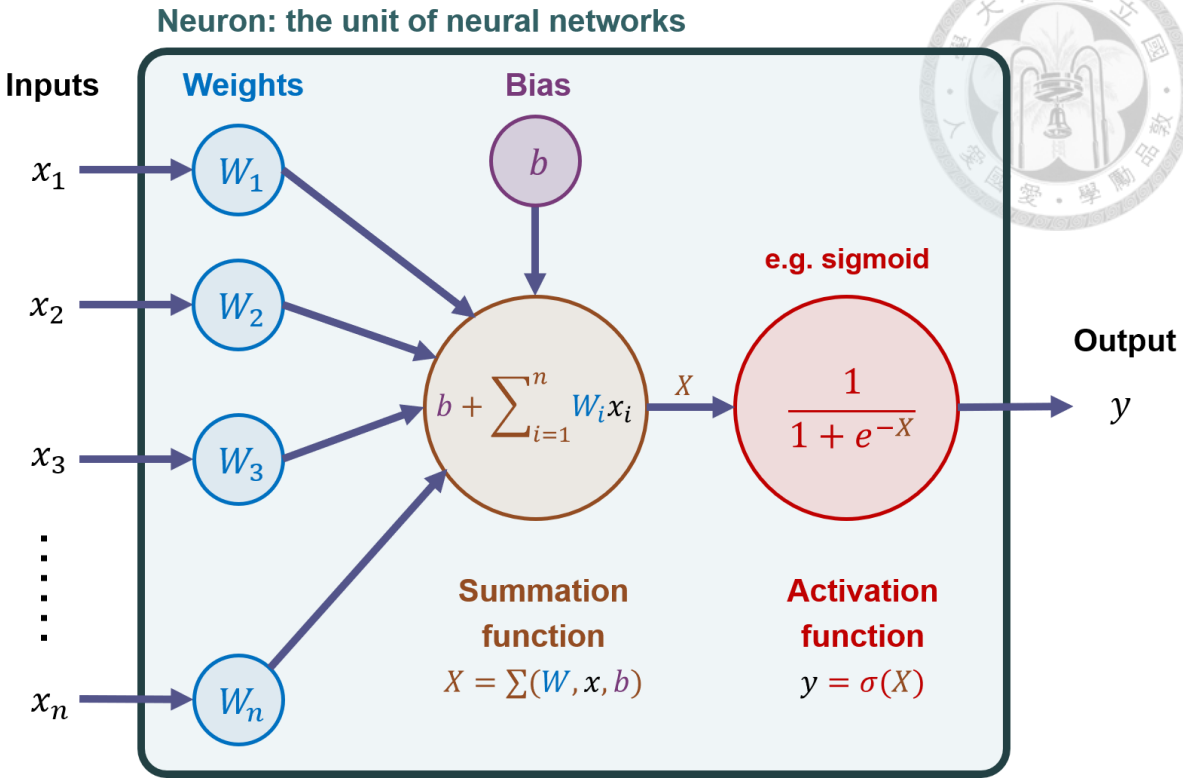


Figure F2. The components of a neuron: weights, bias and activation function.

F.2. RNN-based Chemical Semantic Model

Recurrent neural networks (RNNs) are a family of neural networks specialized for processing a sequential data. They are widely used in the context generation in response to input texts, as known as the sequence-to-sequence tasks. The applications of RNN models include natural language processing[400, 401], machine translation, chatbot, and musical composition[402]. An RNN network is composed of one or multiple unit. There are two common units: long short-term memory units (LSTMs) and gated recurrent units (GRUs)[397, 403]. As shown by **Figure F3**, either type of the units has a past-memory cell (LSTM: c; GRU: h), a new-memory cell (LSTM: $\tilde{c}$; GRU: $\tilde{h}$), and several gates (LSTM: i, o, and f; GRU: r and z). The overall hidden state of an RNN, denoted as $\boldsymbol{h_t}$, depends on the information stored in memory cells and the states of gates within each unit. The operation of an RNN can be illustrated by **Figure F4**. When the RNN is

doi:10.6342/NTU202403528

operating in free-running mode, or inference mode, it can be conceptualized as being unfolded into a sequence of time steps that it has processed. In this scenario, the output token generated in each time step serves as the input token for the subsequent time step.

In the training process, a non-numerical sequential data $\boldsymbol{S}$ is decomposed into T tokens $[\boldsymbol{s_1}, \boldsymbol{s_2}, \dots, \boldsymbol{s_t}, \dots, \boldsymbol{s_T}]$ before fed into an RNN, and the unique tokens are stored into a library. Each token $\boldsymbol{s_j}$ in the library is associated with a unique numerical feature vector $\boldsymbol{e_j}$ through entity embedding[401, 404] or one-hot encoding[266, 401]. Based on this, the tokenized data $[\boldsymbol{s_1}, \boldsymbol{s_2}, \dots, \boldsymbol{s_t}, \dots, \boldsymbol{s_T}]$ is transformed into a feature vector form $[\boldsymbol{e_1}, \boldsymbol{e_2}, \dots, \boldsymbol{e_t}, \dots, \boldsymbol{e_T}]$. Instead of token $\boldsymbol{s_t}$, it is the feature vector $\boldsymbol{e_t}$ that the mathematical operations in the RNN are conducted to. When a feature vector $\boldsymbol{e_i}$ is inputted to a unit, the state of gates will update through eq. (F.2–1) to (F.2–3), eq. (F.2–7), and eq. (F.2–8). Subsequently, the feature vector $\boldsymbol{e_t}$ is encoded into the new-memory cell through eq. (F.2–4) and eq. (F.2–9). Then, the contents in the past-memory cell are mixed with the those in the new-memory cell, thereby forming the hidden state $\boldsymbol{h_t}$, as illustrated by eq. (F.2–5), eq. (F.2–6), and eq. (F.2–10).

- **Long short-term memory unit (LSTM):**

$$\boldsymbol{o_t} = \sigma(\boldsymbol{W_o e_t} + \boldsymbol{U_o h_{t-1}} + \boldsymbol{V_o c_t} + \boldsymbol{b_o}) \quad \text{(F.2–1)}$$

$$\boldsymbol{f_t} = \sigma\big(\boldsymbol{W_f e_t} + \boldsymbol{U_f h_{t-1}} + \boldsymbol{V_f c_{t-1}} + \boldsymbol{b_f}\big) \quad \text{(F.2–2)}$$

$$\boldsymbol{i_t} = \sigma(\boldsymbol{W_i e_t} + \boldsymbol{U_i h_{t-1}} + \boldsymbol{V_i c_{t-1}} + \boldsymbol{b_i}) \quad \text{(F.2–3)}$$

$$\tilde{\boldsymbol{c}}_t = \tanh(\boldsymbol{W_c e_t} + \boldsymbol{U_c h_{t-1}} + \boldsymbol{b_c}) \quad \text{(F.2–4)}$$

$$\boldsymbol{c_t} = \boldsymbol{f_t} \odot \boldsymbol{c_{t-1}} + \boldsymbol{i_t} \odot \tilde{\boldsymbol{c}}_t \quad \text{(F.2–5)}$$

$$\boldsymbol{h_t} = \boldsymbol{o_t} \odot \tanh(\boldsymbol{c_t}) \quad \text{(F.2–6)}$$

where σ is the activation functions for the gates. $\boldsymbol{i_t}$, $\boldsymbol{o_t}$, and $\boldsymbol{f_t}$ are the states of "input", "output", and "forget" gates, respectively. The subscript $\boldsymbol{t}$ denotes time step. $\boldsymbol{W_o}$, $\boldsymbol{W_f}$,

doi:10.6342/NTU202403528

$\boldsymbol{W_i}$, $\boldsymbol{W_c}$, $\boldsymbol{U_o}$, $\boldsymbol{U_f}$, $\boldsymbol{U_i}$, $\boldsymbol{U_c}$, $\boldsymbol{V_o}$, $\boldsymbol{V_f}$, and $\boldsymbol{V_i}$, are the weights. $\boldsymbol{h_t}$ is the hidden state. $\odot$ is the entry-wise multiplication operator.

- **Gated recurrent unit (GRU):**

$$\boldsymbol{r_t} = \sigma(\boldsymbol{W_r e_t} + \boldsymbol{U_r h_{t-1}} + \boldsymbol{b_r}) \quad \text{(F.2–7)}$$

$$\boldsymbol{z_t} = \sigma(\boldsymbol{W_z e_t} + \boldsymbol{U_z h_{t-1}} + \boldsymbol{b_z}) \quad \text{(F.2–8)}$$

$$\widetilde{\boldsymbol{h}}_t = \tanh[\boldsymbol{W_h e_t} + \boldsymbol{U_h}(\boldsymbol{r_t} \odot \boldsymbol{h_{t-1}}) + \boldsymbol{b_h}] \quad \text{(F.2–9)}$$

$$\boldsymbol{h_t} = (\boldsymbol{1} - \boldsymbol{z_t}) \odot \boldsymbol{h_{t-1}} + \boldsymbol{z_t} \odot \widetilde{\boldsymbol{h}}_t \quad \text{(F.2–10)}$$

where σ is the activation functions for the gates. $\boldsymbol{r_t}$ and $\boldsymbol{z_t}$ are the states of "reset" and "update" gates, respectively. $\boldsymbol{W_r}$, $\boldsymbol{W_z}$, $\boldsymbol{W}$, $\boldsymbol{U_r}$, $\boldsymbol{U_z}$, and $\boldsymbol{U}$ are the weights. $\boldsymbol{h_t}$ is the hidden state. $\odot$ is the entry-wise multiplication operator, as known as Hadamard product.

Finally, as presented in eq. (F.2–12), a softmax layer is used to compute the probability distribution $\boldsymbol{p_t} = \boldsymbol{p}(\boldsymbol{h_t}; \boldsymbol{\theta}) = [p_{t1}, p_{t2}, \dots, p_{tN}]$ for classifying the $\boldsymbol{h_t}$ as a certain feature vector in the $[\boldsymbol{e_1}, \boldsymbol{e_2}, \dots, \boldsymbol{e_N}]$, where N is the total number of unique tokens in the library, and $\boldsymbol{\theta}$ represents all of the involved weights and bias. Based on the probability distribution $\boldsymbol{p_t}$, a feature vector is sampled from the library as the output $\boldsymbol{e_{t+1}}$, whose corresponding token $\boldsymbol{s_{t+1}}$ can be determined from the token-embedding relations established before, as presented in eq. (F.2–13).

$$\boldsymbol{a_t} = \boldsymbol{W_s h_t} + \boldsymbol{b_s} = [\boldsymbol{W_{s1} h_t} + b_{s1}; \dots; \boldsymbol{W_{sN} h_t} + b_{sN}] \quad \text{(F.2–11)}$$

$$\boldsymbol{p_t} = \boldsymbol{p}(\boldsymbol{h_t}; \boldsymbol{\theta}) = \frac{\exp(\boldsymbol{a_t})}{\sum_{\boldsymbol{j=1}}^{N} \exp(\boldsymbol{W_{sj} h_t} + b_{sj})} \quad \text{(F.2–12)}$$

$$\boldsymbol{s_{t+1}} \leftarrow \boldsymbol{p_t} \quad \text{(F.2–13)}$$

doi:10.6342/NTU202403528

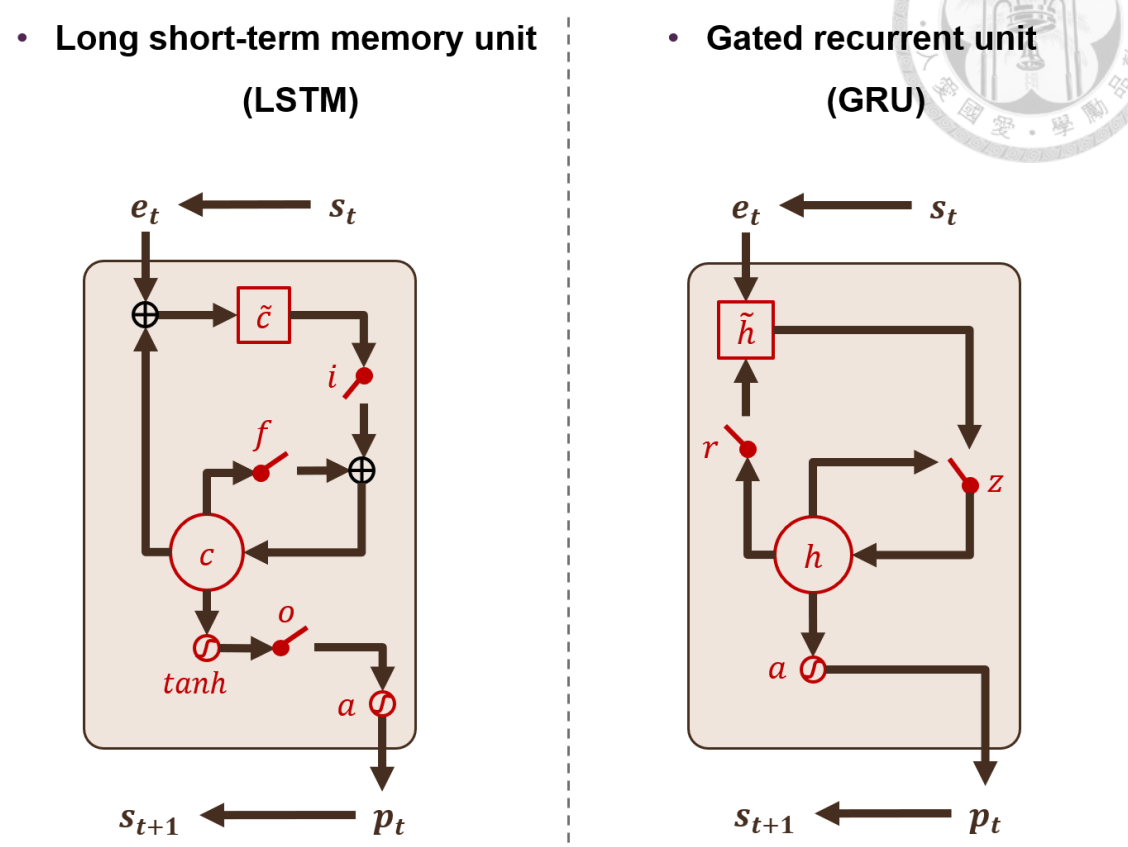


Figure F3. Two types of RNN units: LSTM and GRU. s_t is the input token, e_t is the feature vector for s_t, p_t is the probability distribution for library tokens, e_{t+1} is the feature vector for the output token sampled from p_t, and s_{t+1} is the output token. c and h are past-memory cells, $\tilde{c}$ and $\tilde{h}$ are new-memory cells, i, f, o, r, and z are gates, and a is softmax layer for classification.[403]

doi:10.6342/NTU202403528

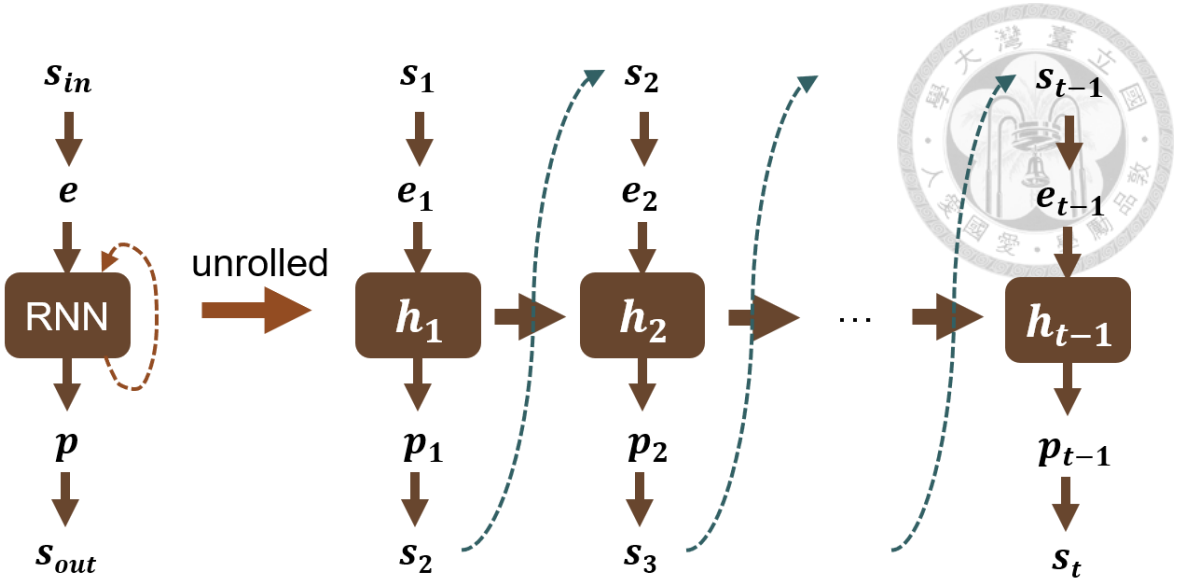


Figure F4. The free-running mode of an RNN in a context-generating task.

Given that the training process is meant to reproduce the training data $\boldsymbol{S} = [\boldsymbol{s_1}, \boldsymbol{s_2}, \ldots, \boldsymbol{s_t}, \ldots, \boldsymbol{s_T}]$, the conditional probability for generating the expected contexts is used to defined as loss function for the RNN. As shown by eq. (F.2–14), the minimization of the loss function $\mathcal{L}_{RNN}$ is equivalent to maximizing the probability to generate $\boldsymbol{S}$ from $\boldsymbol{s_1}$.

$$\mathcal{L}_{RNN} = -\log[p(\boldsymbol{S};\boldsymbol{\theta})] = -\sum_{t=1}^{T} \log[p(\boldsymbol{s_t}|\boldsymbol{s_{t-1}}, \ldots, \boldsymbol{s_1};\boldsymbol{\theta})] \qquad \text{(F.2–14)}$$

The work by M. Olivecrona et al. (2017)[276] and M. H. S. Segler et al. (2018)[277] are two examples for applying RNN to computational molecular design. Their framework of RNN-based molecular design is depicted in **Figure 6.1-1**. In their work, the molecular structures are represented by language-like SMILES (Simplified Molecular-Input Line-Entry System) strings.[74] The semantic significance of a SMILES string relies on the correct arrangements of constituent characters under the grammatical rules.[74] For instances, "C(C(O))" is ethanol, "C(O(C))" is diethyl ether, and "C(C(C)" is an invalid

doi:10.6342/NTU202403528

SMILES string since the left and right round brackets need to be paired for the closure of a chain. To construct a syntactic model capable of generating valid SMILES strings, both research groups gathered approximately 1.5 million SMILES strings from the ChEMBL database. These samples were then utilized to train an LSTM-based RNN model. In the training process the RNN parameters $\boldsymbol{\theta}$ are optimized such that the learning loss reached its minimum value. Subsequently, the trained RNN model served as a generative model for producing new molecules. To ensure the validity of the newly generated SMILES strings, RDKit parsing was employed for examination. Valid SMILES strings were then subjected to property models. The valid chemical species, along with their associated properties, were appended to the training data. Then, a subset of the expanded training data, comprising molecules exhibiting optimal performance properties, was used to retrain the RNN model. This technique, referred to as transfer learning, aimed to enhance the model's specialization for particular specifications in the molecular design task. The iterative process of retraining the model and generating new molecular species is performed until a chemical candidate satisfying the property specifications is identified.

Regarding the performance evaluation of their RNN model, approximately 94% to 98% of the generated SMILES strings are valid. Furthermore, approximately 90% of these valid SMILES strings fall outside the scope of the training data, highlighting the model's ability to generate novel chemical structures. Additionally, about 89% of the valid SMILES strings correspond to unique chemical species. Despite of these promising metrics, there are several limitations on the applicability of RNN. Firstly, it is important to note that, due to architectural nature, training or running an RNN model can demand much higher computational resources (e.g. memory bandwidth) compared to convolutional and linear neural network layers.[405] Secondly, it may be difficult to train LSTM-based and GRU-based RNN models due to the problem of exploding gradients.[406,

doi:10.6342/NTU202403528

[407] This problem is caused by the dynamical behavior of serial mathematical operations by NN layers. Take LSTM-based RNN as the example. To see this issue, let us write down the gradients of $\mathcal{L}_{RNN(LSTM)}$ with respect to a particular weight $\boldsymbol{W}$ in the neural network, as presented in eq. (F.2–15).[408] In this equation, $\boldsymbol{a_t}$ is referred to eq. (F.2–11), and $\boldsymbol{h_t}$ is referred to eq. (F.2–6) .

$$\boldsymbol{\nabla_W \mathcal{L}_{RNN(LSTM)}} = -\sum_{t=1}^{T} \frac{\partial \log[p(\boldsymbol{s_t}|\boldsymbol{s_{t-1}}, \dots, \boldsymbol{s_1})]}{\partial \boldsymbol{a_t}} \frac{\partial \boldsymbol{a_t}}{\partial \boldsymbol{h_t}} \frac{\partial \boldsymbol{h_t}}{\partial \boldsymbol{c_t}} \frac{\partial \boldsymbol{c_t}}{\partial \boldsymbol{c_1}} \frac{\partial \boldsymbol{c_1}}{\partial \boldsymbol{W}} \tag{F.2–15}$$

In particular, the chain rule in eq. (F.2–15) involves the derivatives of memory states from the time step 1 to t, as explicitly expressed by eq. (F.2–16). Substituting the variables in eq. (F.2–16) with eq. (F.2–2), eq. (F.2–3), and eq. (F.2–5) results in the full expression, as shown by eq. (F.2–17).

$$\frac{\partial \boldsymbol{c_t}}{\partial \boldsymbol{c_1}} = \frac{\partial \boldsymbol{c_t}}{\partial \boldsymbol{c_{t-1}}} \frac{\partial \boldsymbol{c_{t-1}}}{\partial \boldsymbol{c_{t-2}}} \dots \frac{\partial \boldsymbol{c_2}}{\partial \boldsymbol{c_1}} \tag{F.2–16}$$

$$\begin{aligned} \frac{\partial \boldsymbol{c_t}}{\partial \boldsymbol{c_1}} = \prod_{k=2}^{t} \Big[\boldsymbol{f_k} + \boldsymbol{V_f} \big[\sigma' \big(\boldsymbol{W_f e_k} + \boldsymbol{U_f h_{k-1}} + \boldsymbol{V_f c_{k-1}} + \boldsymbol{b_f} \big) \odot \boldsymbol{c_{k-1}} \big] \\ + \boldsymbol{V_i} [\sigma' (\boldsymbol{W_i e_k} + \boldsymbol{U_i h_{k-1}} + \boldsymbol{V_i c_{k-1}} + \boldsymbol{b_i}) \odot \tilde{\boldsymbol{c}}_k] \Big] \end{aligned} \tag{F.2–17}$$

It should be noted that the activation function σ is specified by user before training and its mathematical form remains unchanged during the training process. Therefore, it is relatively easy to regulate the derivative of activation function σ' and the state of forget gate $\boldsymbol{f_k}$ to avoid the exploding gradients. However, the optimal $\boldsymbol{V_f}$ and $\boldsymbol{V_i}$ are undetermined until the training finishes successfully, hence they are typically the predominant factors in eq. (F.2–17), especially when $t \gg 1$. Suppose $\boldsymbol{V_i}$ term is more

doi:10.6342/NTU202403528

predominant than $\boldsymbol{V_f}$ term, then we have eq. (F.2–18).[406, 407]

$$\left\|\frac{\partial \boldsymbol{c_t}}{\partial \boldsymbol{c_1}}\right\| \approx \prod_{k=2}^{t} \|\boldsymbol{V_i}\| \|\text{diag}(\sigma'(\boldsymbol{W_i e_k} + \boldsymbol{U_i h_{k-1}} + \boldsymbol{V_i c_{k-1}} + \boldsymbol{b_i}) \odot \tilde{\boldsymbol{c}}_k)\| \quad \text{(F.2–18)}$$

From eq. (F.2–2) and eq. (F.2–5), we know that $\boldsymbol{V_i}$ would be a $n \times n$ square matrix if both the $\tilde{\boldsymbol{c}}$ vector and $\boldsymbol{c}$ vector have dimension of n. The eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ and the corresponding eigenvectors $\boldsymbol{q_1}, \boldsymbol{q_2}, \ldots, \boldsymbol{q_n}$ of matrix $\boldsymbol{V_i}$ can be obtained through eigen-decomposition[406, 407].

$$\boldsymbol{V_i} = \boldsymbol{Q D Q^{-1}} \quad \text{(F.2–19)}$$

$$\boldsymbol{D} = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n) \quad \text{(F.2–20)}$$

$$\boldsymbol{Q} = [\boldsymbol{q_1}, \boldsymbol{q_2}, \ldots, \boldsymbol{q_n}] \quad \text{(F.2–21)}$$

If the largest eigenvalue λ_1 is smaller than 1, the derivatives of memory states, i.e. eq. (F.2–16), will vanish. Consequently, the gradient of loss function $\boldsymbol{\nabla_W \mathcal{L}_{RNN(LSTM)}}$ and parameter update $\boldsymbol{\Delta\theta}$ also vanish based on eq. (F.2–15) and eq. (F.1–3) respectively. As the reset gate $\boldsymbol{r_t}$ in GRU and the forget gate $\boldsymbol{f_t}$ in LSTM (see eq. (F.2–17)) will contribute to the derivatives of memory states, both units are usually free from the problem of vanishing gradients[407, 408],.

On the other hand, if the value of $\boldsymbol{\lambda_1}$ is greater than 1, the derivatives of memory states will explode. Consequently, the gradient of loss function $\boldsymbol{\nabla_W \mathcal{L}_{RNN(LSTM)}}$ and parameter update $\boldsymbol{\Delta\theta}$ also explode. Additional mechanisms need to be employed for improvement, though they would increase the architectural and technical complexity of the model.[407, 409] For example, the "teacher forcing" technique suggests that the exploding

doi:10.6342/NTU202403528

gradients in the training process can be prevented by always using the ground-truth token as the input for the next time step[34, 409, 410]. It has been found that the RNN-based molecular design without "teacher forcing" would generate nearly 0% valid SMILES strings.[34]

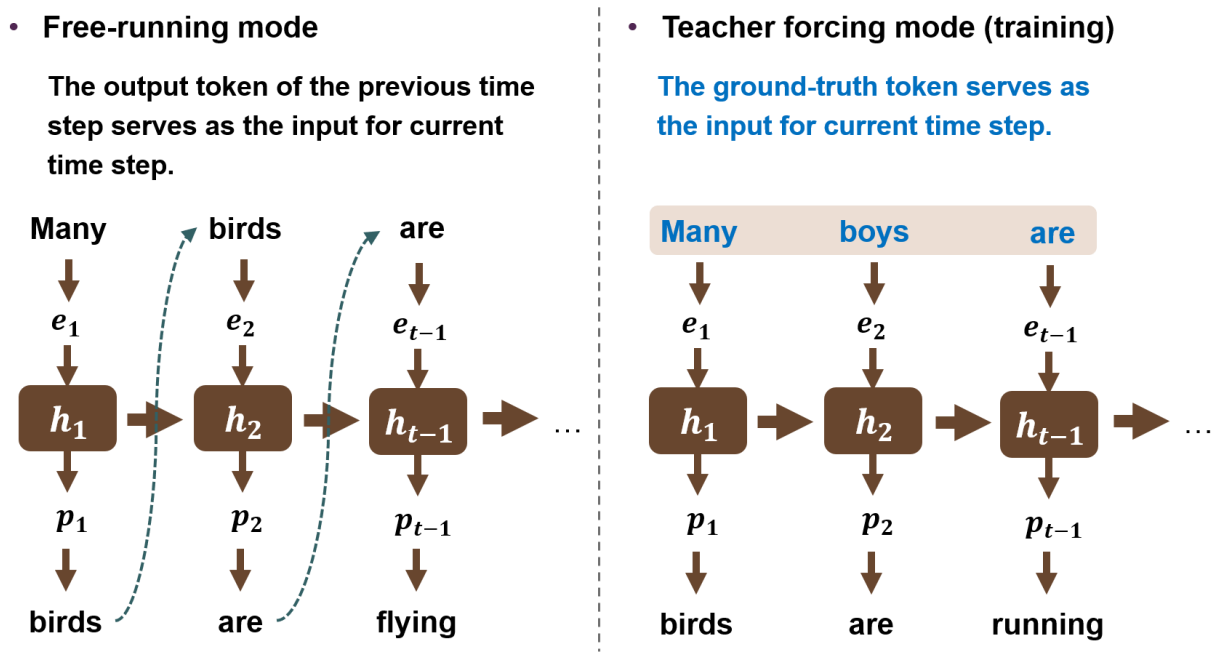


Figure F5. Comparison between the free-running mode and the teacher forcing mode of RNN.

F.3. VAE-based Latent Variable Model

Plain autoencoders (AEs) and variational autoencoders (VAEs) are devised to represent a high-dimension data by a lower-dimension vector, thereby compressing the data into a more compact format. These autoencoders have been proven to be useful in tasks such as translation, drug design, and image processing. The architectures for the two types of autoencoders are shown by **Figure F6**.

doi:10.6342/NTU202403528

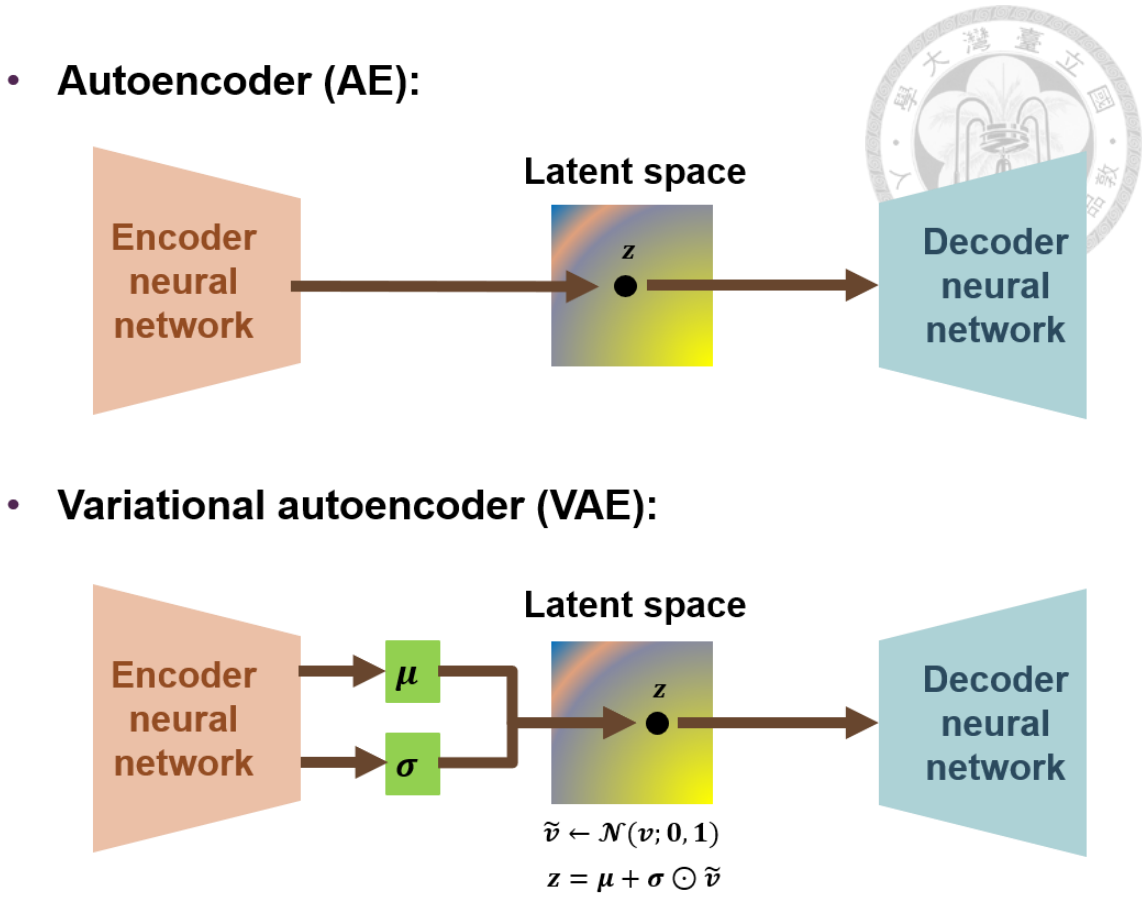


Figure F6. The architectures of plain autoencoders (AEs) and variational autoencoders (VAEs)

Both types of the autoencoders consists of an encoder and a decoder, each of which is a neural network. The encoder extracts the feature patterns of the input data and maps them to a relatively low-dimension vector space called latent space. The decoder recovers a data to its original format from the limited features recorded in its latent space representation. The primary mechanistic distinction between the two types of autoencoders lies in their respective approaches to determining the latent space representation. The VAEs map an encoder output to the mean value $\boldsymbol{\mu} = [\mu_1, \dots, \mu_i, \dots]$ and the standard deviation $\boldsymbol{\sigma} = [\sigma_1, \dots, \sigma_i, \dots]$ of a multi-dimensional normal distribution, i.e. eq. (F.3–1). In other words, VAEs encode each input data point into a probability distribution, thus exhibiting stochastic features. In contrast, plain AEs directly map an

doi:10.6342/NTU202403528

encoder output to a single fixed point in the latent space, making them deterministic in nature.

$$\mathcal{N}(z_i;\mu_i,\sigma_i^2)=\frac{1}{\sigma_i\sqrt{2\pi}}\exp\left[-\frac{1}{2}\left(\frac{z_i-\mu_i}{\sigma_i}\right)^2\right] \tag{F.3–1}$$

In VAEs, the latent space representation of, denoted as $\boldsymbol{z}=[z_1,\dots,z_i,\dots]$, is determined by sampling from a modified distribution rather than eq. (F.3–1). In practice, the values sampled by VAEs are actually standard scores v_i, as depicted in eq. (F.3–2). Moreover, the normal distribution of eq. (F.3–1) is approximated by the standard normal distribution $\mathcal{N}(v_i;0,1)$, as demonstrated in eq. (F.3–3).

The sampling process involves two steps. Firstly, a standard score $\tilde{v}_i$ is sampled from $\mathcal{N}(v_i;0,1)$, as illustrated by eq. (F.3–4). Next, multi-dimensional $\tilde{\boldsymbol{v}}=[\tilde{v}_1,\dots,\tilde{v}_i,\dots]$ are transformed into $\boldsymbol{z}$ using eq. (F.3–2), which can be rewritten as a concise vector form using the entry-wise multiplication operator "$\odot$", as presented by eq. (F.3–5). The approximation by eq. (F.3–3) decouples v_i from pre-exponential σ_i and facilitates the update of neural network parameters during the backpropagation step, wherein the gradients of the loss function with respect to $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$, i.e. $\boldsymbol{\nabla_{\mu}}\mathcal{L}_{VAE}$ and $\boldsymbol{\nabla_{\sigma}}\mathcal{L}_{VAE}$, are computed. Since in eq. (F.3–5) the stochastic nature is factored out into the $\tilde{\boldsymbol{v}}$ term, the two aforementioned gradients become clearly-defined and computable as $\tilde{\boldsymbol{v}}$ is treated as a constant per sample.[411]

$$v_i=\frac{z_i-\mu_i}{\sigma_i} \tag{F.3–2}$$

$$\mathcal{N}(v_i;\mu_i,\sigma_i^2)=\frac{1}{\sigma_i\sqrt{2\pi}}\exp\left(-\frac{v_i^2}{2}\right)=\frac{\mathcal{N}(v_i;0,1)}{\sigma_i}\sim\mathcal{N}(v_i;0,1) \tag{F.3–3}$$

doi:10.6342/NTU202403528

$$\tilde{v}_i \leftarrow \mathcal{N}(v_i; 0,1) \tag{F.3–4}$$

$$\boldsymbol{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \tilde{\boldsymbol{v}} \tag{F.3–5}$$

In training step, the optimizer seeks the best model parameters by minimization of loss function. Therefore, the distinction between plain AEs and VAEs also characterized by their loss function. Let $\boldsymbol{x}$ be the input data, $\mathcal{D}$ be the decoder function, $\mathcal{E}$ be the encoder function, KL be the Kullback-Leibler divergence, $\mathcal{N}(\boldsymbol{v}; \boldsymbol{\mu}, \boldsymbol{\sigma^2})$ be the multi-dimensional normal distribution, and $\boldsymbol{z}$ be the latent space representation for a data in VAE. Then, the loss function for AEs is written as eq. (F.3–6), and the loss function for VAEs is written as eq. (F.3–7).

$$\mathcal{L}_{AE} = |\boldsymbol{x} - \mathcal{D}(\mathcal{E}(\boldsymbol{x}))|^2 \tag{F.3–6}$$

$$\mathcal{L}_{VAE} = |\boldsymbol{x} - \mathcal{D}(\boldsymbol{z})|^2 + KL[\mathcal{N}(\boldsymbol{v}; \boldsymbol{0}, \boldsymbol{1}), \mathcal{N}(\boldsymbol{v}; \boldsymbol{\mu}, \boldsymbol{\sigma^2})] \tag{F.3–7}$$

$$KL[\mathcal{N}(\boldsymbol{v}; \boldsymbol{0}, \boldsymbol{1}), \mathcal{N}(\boldsymbol{v}; \boldsymbol{\mu}, \boldsymbol{\sigma^2})] = -\int_{v_i \to -\infty}^{v_i \to \infty} \mathcal{N}(\boldsymbol{v}; \boldsymbol{0}, \boldsymbol{1}) \ln \frac{\mathcal{N}(\boldsymbol{v}; \boldsymbol{\mu}, \boldsymbol{\sigma^2})}{\mathcal{N}(\boldsymbol{v}; \boldsymbol{0}, \boldsymbol{1})} d\boldsymbol{v} \tag{F.3–8}$$

Both of the loss functions have a reconstruction loss as their first term in the right-hand side, which characterizes the reconstruction rate of an autoencoder, i.e. the success rate to restore input data $\boldsymbol{x}$ after it goes through encoder, latent space, and decoder. The loss function of VAEs have an extra KL divergence term, which is a measure for the dissimilarity between two probability distributions, as shown by eq. (F.3–8). By incorporating the KL divergence term in the loss function, the learning process encourages $\mathcal{N}(\boldsymbol{v}; \boldsymbol{\mu}, \boldsymbol{\sigma^2})$ to be progressively asymptotic to $\mathcal{N}(\boldsymbol{v}; \boldsymbol{0}, \boldsymbol{1})$ during the learning steps. Consequently, VAEs can form a relatively small-range and compact distribution around the origin of latent space, as demonstrated by **Figure F7**.[411]

doi:10.6342/NTU202403528

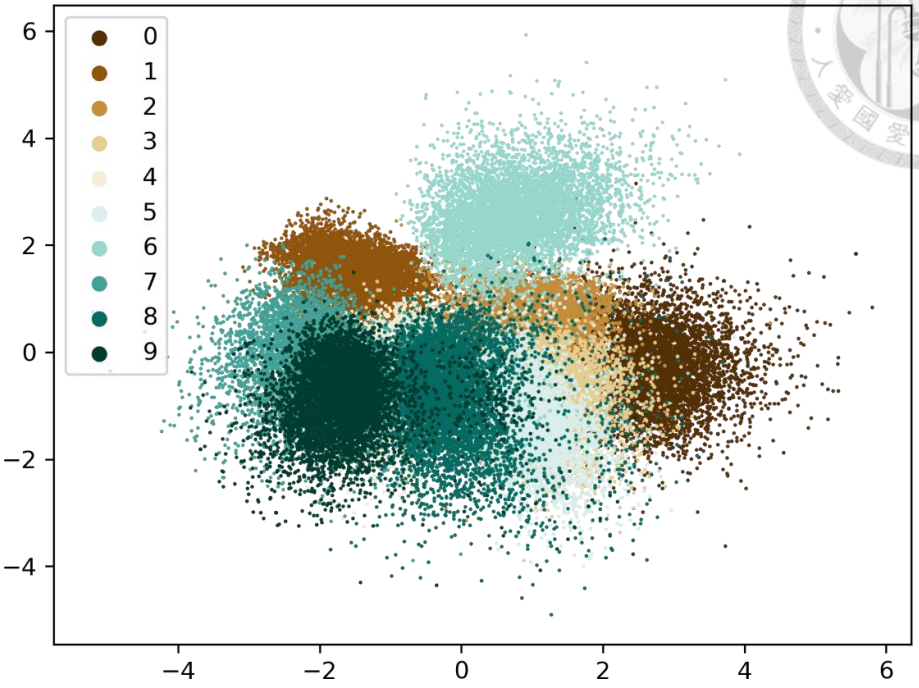


Figure F7. A typical latent space distribution for variational autoencoders (VAEs).[411]

On the other hand, it has been pointed out that plain AEs may encounter robustness issues when used as generative models. For a plain AE, the heterogeneous training data tend to form different large-scale sparse clusters in the latent space, as demonstrated by **Figure F8**.[411] As the plain AE-based generative models rely on the sampling in latent space for new data, the unpopulated region in latent space can result in ineligible data.

doi:10.6342/NTU202403528

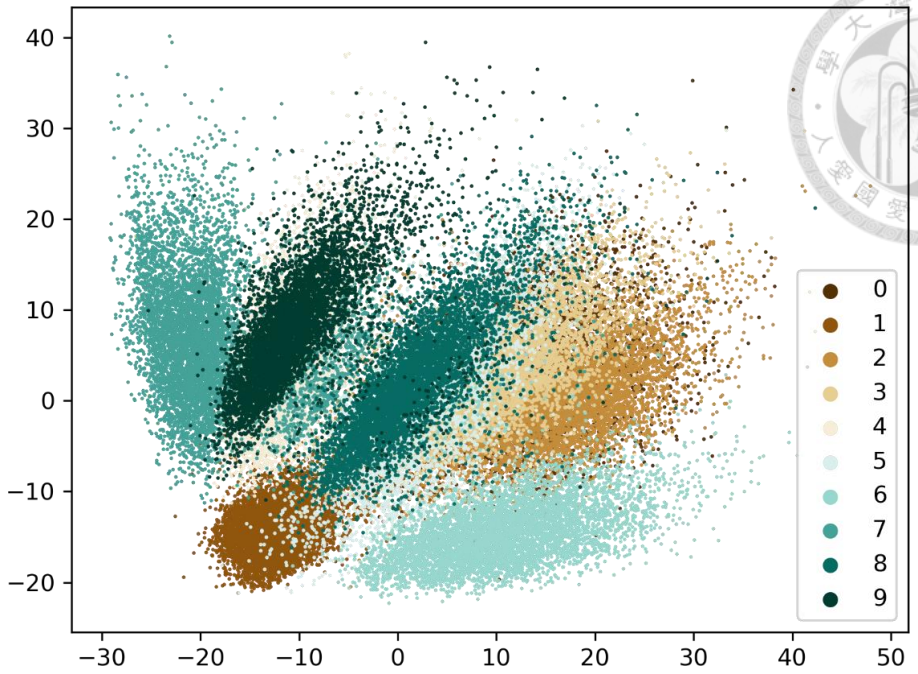


Figure F8. A typical latent space distribution for plain autoencoders (AEs).[411]

R. Gómez-Bombarelli et al. (2018)[281] and S. Mohammadi et al. (2019)[412] are two pioneering works that utilize the VAEs for molecular design. As illustrated by **Figure 6.1-2**, the typical architecture of VAE-based molecular design consists of a VAE and neural-network based (NN-based) property models. Their training data for the VAE are chemical structures in SMILES[74] format. In the training process, one feeds a vast number of chemical structures to encoder and requires the decoder to reconstruct those structures from their latent space representation. Meanwhile, the KL divergence term in the loss function helps the encoder to create relatively continuous and smooth clusters in latent space. As a chemical structure is encoded into a point in latent space, new chemical species can be generated by a sampling of new points or a redistribution of existing points in the latent space. These new points are then sent to NN-based QSPR models for prediction of properties, and outputted into readable chemical structures by the decoder.

doi:10.6342/NTU202403528

F.4. Transformer Architecture with Self-Attention Mechanism

The integration of transformer architecture with self-attention mechanisms[280, 311] into VAE- and RNN-based generative models has significantly improved model performance. Notably, chemical validity and reconstruction rates have been elevated to 98-99.9%[280, 284, 291, 312], and chemical novelty and uniqueness have surpassed 80%[280, 284].

Key advantages of transformers over RNNs lie in the self-attention mechanism and the presence of multiple "attention heads". Attention heads capture different semantic aspects of the input sequence, enabling the model to capture comprehensive information. The self-attention mechanism effectively addresses the long-range dependency problem inherent in RNNs, where correlations between distant tokens often diminish during training due to the vanishing gradient issue (section F.2). The self-attention mechanism transforms the input embedding matrix $\boldsymbol{E_m}$ into query $\boldsymbol{Q_i}$. key $\boldsymbol{K_i}$, and value $\boldsymbol{V_i}$ matrices. These matrices are subsequently employed to compute attention scores for each head, $\mathbf{head_i}$. The heads are concatenated to determine the next output token.

$$\boldsymbol{E_m} = [\boldsymbol{a_1}; \boldsymbol{a_2}; \dots; \boldsymbol{a_m}] \tag{F.4–1}$$

$$\boldsymbol{E_t} = [\boldsymbol{b_1}; \boldsymbol{b_2}; \dots; \boldsymbol{b_t}] \tag{F.4–2}$$

$$\boldsymbol{V_i} = \left[\boldsymbol{v_{1,i}}; \boldsymbol{v_{2,i}}; \dots; \boldsymbol{v_{m,i}}\right] = \boldsymbol{E_m W_i^V} \tag{F.4–3}$$

$$\boldsymbol{K_i} = \left[\boldsymbol{k_{1,i}}; \boldsymbol{k_{2,i}}; \dots; \boldsymbol{k_{m,i}}\right] = \boldsymbol{E_m W_i^K} \tag{F.4–4}$$

$$\boldsymbol{Q_i} = \left[\boldsymbol{q_{1,i}}; \boldsymbol{q_{2,i}}; \dots; \boldsymbol{q_{t,i}}\right] = \boldsymbol{E_t W_i^Q} \tag{F.4–5}$$

$$\mathbf{head_i} = \text{attention}(\boldsymbol{Q_i}, \boldsymbol{K_i}, \boldsymbol{V_i}; d_k) = \text{softmax}\left(\frac{\boldsymbol{Q_i K_i^T} + \boldsymbol{M}}{\sqrt{d_k}}\right)\boldsymbol{V_i} \tag{F.4–6}$$

$$\boldsymbol{O_p} = [\boldsymbol{o_1}; \boldsymbol{o_2}; \dots; \boldsymbol{o_p}] = \text{horizontal\_concat}(\mathbf{head_1}, \dots, \mathbf{head_h})\boldsymbol{W^O} \tag{F.4–7}$$

doi:10.6342/NTU202403528

$$\boldsymbol{P_t} = [\boldsymbol{p_1}; \boldsymbol{p_2}; \ldots; \boldsymbol{p_t}] = \text{softmax}(\boldsymbol{O_t W^P}) \quad \text{(F.4–8)}$$

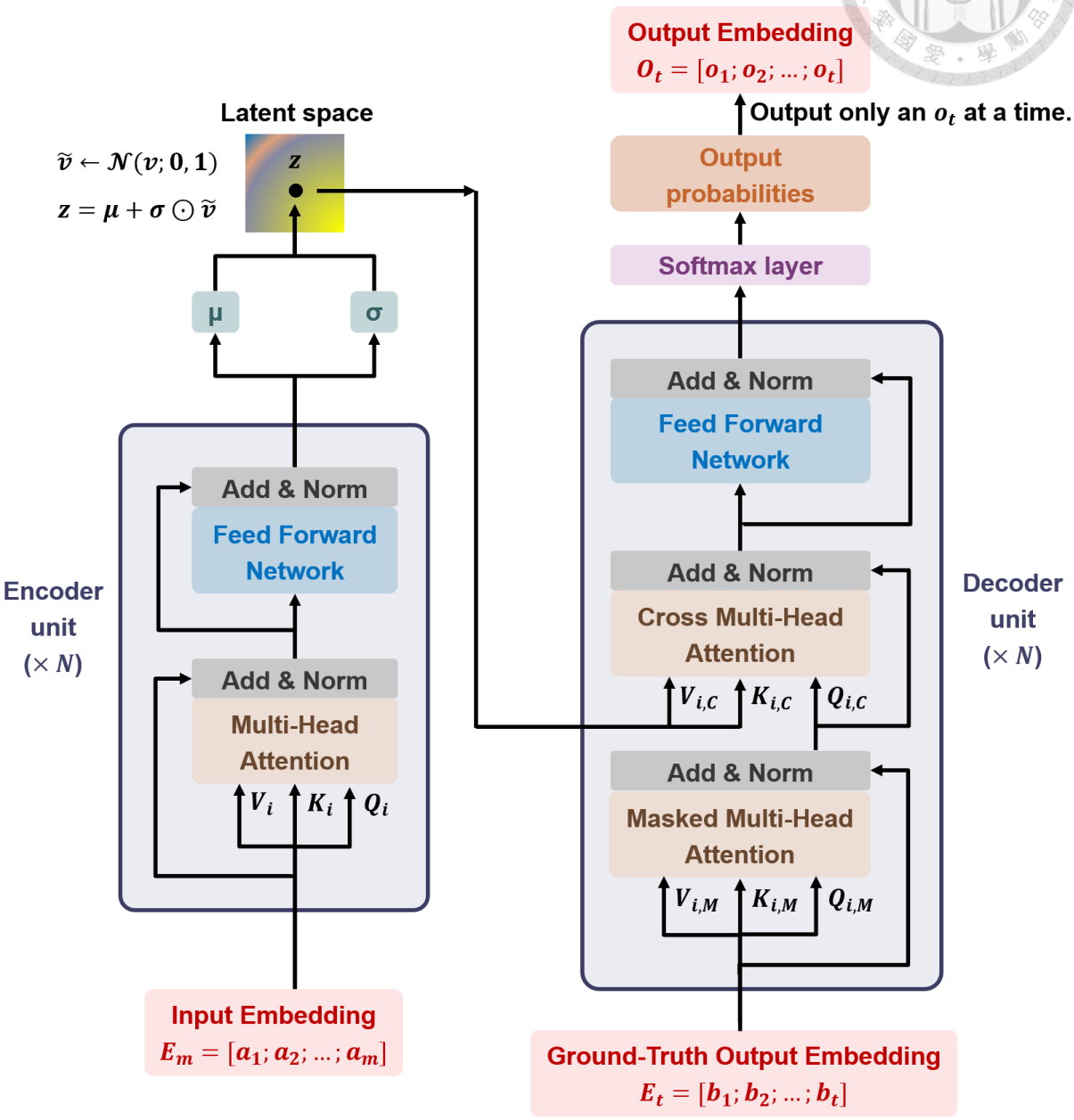


Figure F9. Transformer-based VAE: training mode.

doi:10.6342/NTU202403528

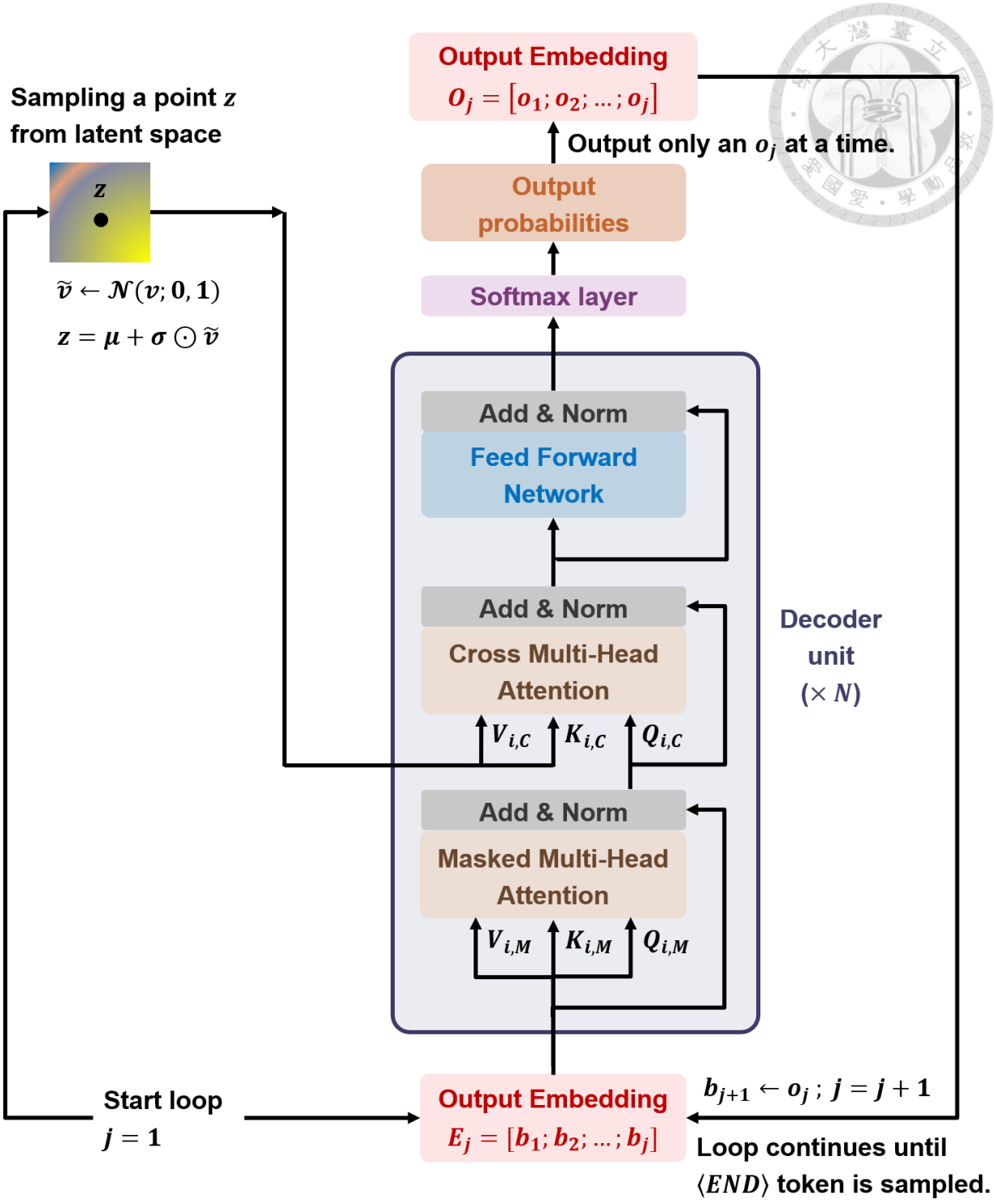


Figure F10. Transformer-based VAE: free running mode.

doi:10.6342/NTU202403528

References

1. Pollak, P., *Fine chemicals: the industry and the business*. John Wiley & Sons: 2011.
2. Lee, J.-C.; Chai, J.-D.; Lin, S.-T., Assessment of density functional methods for exciton binding energies and related optoelectronic properties. *RSC Advances* **2015,** 5, (123), 101370-101376.
3. Gerbaud, V.; Rodriguez-Donis, I.; Hegely, L.; Lang, P.; Denes, F.; You, X., Review of extractive distillation. Process design, operation, optimization and control. *Chemical Engineering Research and Design* **2019,** 141, 229-271.
4. Haregewoin, A. M.; Wotango, A. S.; Hwang, B.-J., Electrolyte additives for lithium ion battery electrodes: progress and perspectives. *Energy & Environmental Science* **2016,** 9, (6), 1955-1988.
5. Research, G. V. *Specialty Chemicals Market Size, Share & Trends Analysis Report By Product (Institutional & Industrial Cleaners, Flavor & Fragrances, Food & Feed Additives), By Region, And Segment Forecasts, 2020 - 2027*; July, 2020; p 250.
6. Mousavi, S.; Zare, S.; Mirzaei, M.; Feizi, A., Novel Drug Design for Treatment of COVID-19: A Systematic Review of Preclinical Studies. *Canadian Journal of Infectious Diseases and Medical Microbiology* **2022,** 2022, 2044282.
7. Liu, C.; Li, F.; Ma, L.-P.; Cheng, H.-M., Advanced Materials for Energy Storage. *Advanced Materials* **2010,** 22, (8), E28-E62.
8. Wilberforce, T.; Baroutaji, A.; Soudan, B.; Al-Alami, A. H.; Olabi, A. G., Outlook of carbon capture technology and challenges. *Science of The Total Environment* **2019,** 657, 56-72.
9. Santagate, J. *Improving the Product Innovation Process in the Chemicals Industry Through Data Access, Collaboration, and Visibility* International Data Corporation (IDC): Mar., 2016.
10. Miremadi, M.; Musso, C.; Oxgaard, J. *Chemical innovation: An investment for the ages*; McKinsey & Company: May 1, 2013.
11. ChemADVISOR® Online. In UL Solutions: 2023.
12. Lawson, A., The Beilstein Database. In 2008; pp 608-628.
13. ChemSpider | Search and share chemistry. http://www.chemspider.com/

doi:10.6342/NTU202403528

14. Kim, S.; Chen, J.; Cheng, T.; Gindulyte, A.; He, J.; He, S.; Li, Q.; Shoemaker, B. A.; Thiessen, P. A.; Yu, B.; Zaslavsky, L.; Zhang, J.; Bolton, E. E., PubChem 2023 update. *Nucleic Acids Research* **2023,** 51, (D1), D1373-D1380.

15. Ramakrishnan, R.; Dral, P. O.; Rupp, M.; von Lilienfeld, O. A., Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data* **2014,** 1, (1), 140022.

16. Blum, L. C.; Reymond, J.-L., 970 Million Druglike Small Molecules for Virtual Screening in the Chemical Universe Database GDB-13. *Journal of the American Chemical Society* **2009,** 131, (25), 8732-8733.

17. Fink, T.; Reymond, J.-L., Virtual Exploration of the Chemical Universe up to 11 Atoms of C, N, O, F: Assembly of 26.4 Million Structures (110.9 Million Stereoisomers) and Analysis for New Ring Systems, Stereochemistry, Physicochemical Properties, Compound Classes, and Drug Discovery. *Journal of Chemical Information and Modeling* **2007,** 47, (2), 342-353.

18. Ruddigkeit, L.; van Deursen, R.; Blum, L. C.; Reymond, J.-L., Enumeration of 166 Billion Organic Small Molecules in the Chemical Universe Database GDB-17. *Journal of Chemical Information and Modeling* **2012,** 52, (11), 2864-2875.

19. Onken, U.; Rarey-Nies, J.; Gmehling, J., The Dortmund Data Bank: A computerized system for retrieval, correlation, and prediction of thermodynamic properties of mixtures. *International Journal of Thermophysics* **1989,** 10, (3), 739-747.

20. Dortmund Data Bank (DDB). In 2023 ed.; DDBST Dortmund Data Bank Software & Separation Technology GmbH: 2023.

21. Todeschini, R.; Consonni, V., *Molecular descriptors for chemoinformatics: volume I: alphabetical listing/volume II: appendices, references*. John Wiley & Sons: 2009.

22. Fabian, W. M. F., Accurate thermochemistry from quantum chemical calculations? *Monatshefte für Chemie - Chemical Monthly* **2008,** 139, (4), 309-318.

23. Ungerer, P.; Nieto-Draghi, C.; Rousseau, B.; Ahunbay, G.; Lachet, V., Molecular simulation of the thermophysical properties of fluids: From understanding toward quantitative predictions. *Journal of Molecular Liquids* **2007,** 134, (1), 71-89.

24. Zhan, C.-G.; Nichols, J. A.; Dixon, D. A., Ionization Potential, Electron Affinity, Electronegativity, Hardness, and Electron Excitation Energy: Molecular Properties from Density Functional Theory Orbital Energies. *The Journal of Physical Chemistry A* **2003,**

doi:10.6342/NTU202403528

107, (20), 4184-4195.

25. Coley, C. W., Defining and Exploring Chemical Spaces. *Trends in Chemistry* **2021,** 3, (2), 133-145.

26. Drew, K. L. M.; Baiman, H.; Khwaounjoo, P.; Yu, B.; Reynisson, J., Size estimation of chemical space: how big is it? *Journal of Pharmacy and Pharmacology* **2012,** 64, (4), 490-495.

27. Ogata, K.; Isomura, T.; Yamashita, H.; Kubodera, H., A Quantitative Approach to the Estimation of Chemical Space from a Given Geometry by the Combination of Atomic Species. *Qsar & Combinatorial Science* **2007,** 26, (5), 596-607.

28. Polishchuk, P. G.; Madzhidov, T. I.; Varnek, A., Estimation of the size of drug-like chemical space based on GDB-17 data. *Journal of Computer-Aided Molecular Design* **2013,** 27, (8), 675-679.

29. Gorse, A.-D., Diversity in Medicinal Chemistry Space. *Current Topics in Medicinal Chemistry* **2006,** 6, (1), 3-18.

30. Achenie, L.; Venkatasubramanian, V.; Gani, R., *Computer-Aided Molecular Design: Theory and Practice*. 1st ed ed.; Elsevier: Netherlands, 2003; Vol. 12.

31. Doucet, J.-P.; Weber, J., *Computer-Aided Molecular Design: Theory and Applications*. 1st ed.; Academic Press: San Diego, CA 92101, 1996.

32. Austin, N. D.; Sahinidis, N. V.; Trahan, D. W., Computer-aided molecular design: An introduction and review of tools, applications, and solution techniques. *Chemical Engineering Research and Design* **2016,** 116, 2-26.

33. Cheng, Y.; Gong, Y.; Liu, Y.; Song, B.; Zou, Q., Molecular design in drug discovery: a comprehensive review of deep generative models. *Briefings in Bioinformatics* **2021,** 22, (6), bbab344.

34. Elton, D. C.; Boukouvalas, Z.; Fuge, M. D.; Chung, P. W., Deep learning for molecular design—a review of the state of the art. *Molecular Systems Design & Engineering* **2019,** 4, (4), 828-849.

35. Reker, D.; Schneider, G., Active-learning strategies in computer-assisted drug discovery. *Drug Discovery Today* **2015,** 20, (4), 458-465.

36. Deng, J.; Yang, Z.; Ojima, I.; Samaras, D.; Wang, F., Artificial intelligence in drug discovery: applications and techniques. *Briefings in Bioinformatics* **2022,** 23, (1),

doi:10.6342/NTU202403528

bbab430.

37. Floudas, C. A., *Nonlinear and mixed-integer optimization: fundamentals and applications*. Oxford University Press: 1995.

38. Gani, R.; Brignole, E. A., Molecular design of solvents for liquid extraction based on UNIFAC. *Fluid Phase Equilibria* **1983,** 13, 331-340.

39. Joback, K. G. Designing molecules possessing desired physical property values. Dissertation, Massachusetts Institute of Technology, 1989.

40. Odele, O.; Macchietto, S., Computer Aided Molecular Design: A Novel Method for Optimal Solvent Selection. *Fluid Phase Equilib.* **1993,** 82, (Supplement C), 47-54.

41. Achenie, L.; Venkatasubramanian, V.; Gani, R., *Computer-Aided Molecular Design : Theory and Practice*. 1st ed ed.; Elsevier: Netherlands, 2003; Vol. 12.

42. Schneider, G.; Fechner, U., Computer-based de novo design of drug-like molecules. *Nat. Rev. Drug. Discov.* **2005,** 4, 649.

43. Gelin, B. R., Current Approaches in Computer-Aided Molecular Design. In *Computer-Aided Molecular Design*, Reynolds, C. H.; Holloway, M. K.; Cox, H. K., Eds. American Chemical Society: 1995; pp 1-11.

44. Austin, N. D. Tools for Computer-Aided Molecular and Mixture Design. Dissertation, Carnegie Mellon University, 2017.

45. Kutchukian, P. S.; Shakhnovich, E. I., De novo design: balancing novelty and confined chemical space. *Expert Opinion on Drug Discovery* **2010,** 5, (8), 789-812.

46. Schneider, G.; Fechner, U., Computer-based de novo design of drug-like molecules. *Nature Reviews Drug Discovery* **2005,** 4, (8), 649-663.

47. Reymond, J.-L.; Ruddigkeit, L.; Blum, L.; van Deursen, R., The enumeration of chemical space. *WIREs Computational Molecular Science* **2012,** 2, (5), 717-733.

48. McLeese, S. E.; Eslick, J. C.; Hoffmann, N. J.; Scurto, A. M.; Camarda, K. V., Design of ionic liquids via computational molecular design. *Computers & Chemical Engineering* **2010,** 34, (9), 1476-1480.

49. Daina, A.; Michielin, O.; Zoete, V., SwissADME: a free web tool to evaluate pharmacokinetics, drug-likeness and medicinal chemistry friendliness of small molecules. *Scientific Reports* **2017,** 7, (1), 42717.

50. Dong, J.; Wang, N.-N.; Yao, Z.-J.; Zhang, L.; Cheng, Y.; Ouyang, D.; Lu, A.-P.; Cao, D.-S.,

doi:10.6342/NTU202403528

ADMETlab: a platform for systematic ADMET evaluation based on a comprehensively collected ADMET database. *Journal of Cheminformatics* **2018,** 10, (1), 29.

51. Folić, M.; Adjiman, C. S.; Pistikopoulos, E. N., Design of solvents for optimal reaction rate constants. *Aiche Journal* **2007,** 53, (5), 1240-1256.

52. Sahinidis, N. V.; Tawarmalani, M.; Yu, M., Design of alternative refrigerants via global optimization. *Aiche Journal* **2003,** 49, (7), 1761-1775.

53. Marcoulaki, E. C.; Kokossis, A. C., On the development of novel chemicals using a systematic synthesis approach. Part I. Optimisation framework. *Chemical Engineering Science* **2000,** 55, (13), 2529-2546.

54. Samudra, A. P.; Sahinidis, N. V., Optimization-based framework for computer-aided molecular design. *Aiche Journal* **2013,** 59, (10), 3686-3701.

55. Brown, W. M.; Martin, S.; Rintoul, M. D.; Faulon, J.-L., Designing Novel Polymers with Targeted Properties Using the Signature Molecular Descriptor. *Journal of Chemical Information and Modeling* **2006,** 46, (2), 826-835.

56. Churchwell, C. J.; Rintoul, M. D.; Martin, S.; Visco, D. P.; Kotu, A.; Larson, R. S.; Sillerud, L. O.; Brown, D. C.; Faulon, J.-L., The signature molecular descriptor: 3. Inverse-quantitative structure–activity relationship of ICAM-1 inhibitory peptides. *Journal of Molecular Graphics and Modelling* **2004,** 22, (4), 263-273.

57. Faulon, J.-L.; Churchwell, C. J.; Visco, D. P., The Signature Molecular Descriptor. 2. Enumerating Molecules from Their Extended Valence Sequences. *Journal of Chemical Information and Computer Sciences* **2003,** 43, (3), 721-734.

58. Song, Z.; Hu, X.; Zhou, Y.; Zhou, T.; Qi, Z.; Sundmacher, K., Rational design of double salt ionic liquids as extraction solvents: Separation of thiophene/n-octane as example. *Aiche Journal* **2019,** 65, (8), e16625.

59. Chéron, N.; Jasty, N.; Shakhnovich, E. I., OpenGrowth: An Automated and Rational Algorithm for Finding New Protein Ligands. *Journal of Medicinal Chemistry* **2016,** 59, (9), 4171-4188.

60. Fechner, U.; Schneider, G., Flux (2): Comparison of Molecular Mutation and Crossover Operators for Ligand-Based de Novo Design. *Journal of Chemical Information and Modeling* **2007,** 47, (2), 656-667.

61. Fechner, U.; Schneider, G., Flux (1): A Virtual Synthesis Scheme for Fragment-Based

doi:10.6342/NTU202403528

de Novo Design. *Journal of Chemical Information and Modeling* **2006,** 46, (2), 699-707.

62. Sandler, S., *Chemical, Biochemical, and Engineering Thermodynamics*. 5th ed.; John Wiley & Sons: 2017.

63. Wigh, D. S.; Goodman, J. M.; Lapkin, A. A., A review of molecular representation in the age of machine learning. *WIREs Computational Molecular Science* **2022,** 12, (5), e1603.

64. Landrum, G.; Tosco, P.; Kelley, B.; Ric; Cosgrove, D.; Sriniker; Gedeck; Vianello, R.; NadineSchneider; Kawashima, E.; N, D.; Jones, G.; Dalke, A.; Cole, B.; Swain, M.; Turk, S.; AlexanderSavelyev; Vaucher, A.; Wójcikowski, M.; Take, I.; Probst, D.; Ujihara, K.; Scalfani, V. F.; Godin, G.; Lehtivarjo, J.; Walker, R.; Pahl, A.; Berenger, F.; Jasondbiggs; strets *rdkit/rdkit: 2023_03_3 (Q1 2023) Release*, Release_2023_03_3; Zenodo: 2023.

65. O'Boyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T.; Hutchison, G. R., Open Babel: An open chemical toolbox. *Journal of Cheminformatics* **2011,** 3, (1), 33.

66. Norman, N. C.; Pringle, P. G., Hypervalence: A Useful Concept or One That Should Be Gracefully Retired? In *Chemistry*, 2022; Vol. 4, pp 1226-1249.

67. Ahmad, W.-Y.; Omar, S., Drawing Lewis structures: A step-by-step approach. *Journal of Chemical Education* **1992,** 69, (10), 791.

68. Rao, S. S., *Engineering optimization: theory and practice*. John Wiley & Sons: 2019.

69. Sahinidis, N. V.; Grossmann, I. E., Convergence properties of generalized benders decomposition. *Computers & Chemical Engineering* **1991,** 15, (7), 481-491.

70. Geoffrion, A. M., Generalized Benders decomposition. *Journal of Optimization Theory and Applications* **1972,** 10, (4), 237-260.

71. Warr, W. A., Representation of chemical structures. *WIREs Computational Molecular Science* **2011,** 1, (4), 557-579.

72. Faulon, J.-L.; Bender, A., *Handbook of Cheminformatics Algorithms*. Chapman and Hall/CRC: 2010.

73. Hanson, R. M., Jmol SMILES and Jmol SMARTS: specifications and applications. *Journal of Cheminformatics* **2016,** 8, (1), 50.

74. Apodaca, R.; O'Boyle, N.; Dalke, A.; Drie, J. v.; Ertl, P.; Hutchison, G.; James, C. A.; Landrum, G.; Morley, C.; Willighagen, E.; Winter, H. D.; Vandermeersch, T.; May, J. OpenSMILES specification. http://opensmiles.org/opensmiles.html

doi:10.6342/NTU202403528

75. Heller, S. R.; McNaught, A.; Pletnev, I.; Stein, S.; Tchekhovskoi, D., InChI, the IUPAC International Chemical Identifier. *Journal of Cheminformatics* **2015,** 7, (1), 23.
76. Krenn, M.; Häse, F.; Nigam, A.; Friederich, P.; Aspuru-Guzik, A., Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Machine Learning: Science and Technology* **2020,** 1, (4), 045024.
77. Reveil, M.; Clancy, P., Classification of spatially resolved molecular fingerprints for machine learning applications and development of a codebase for their implementation. *Molecular Systems Design & Engineering* **2018,** 3, (3), 431-441.
78. Rogers, D.; Hahn, M., Extended-Connectivity Fingerprints. *Journal of Chemical Information and Modeling* **2010,** 50, (5), 742-754.
79. Capecchi, A.; Probst, D.; Reymond, J.-L., One molecular fingerprint to rule them all: drugs, biomolecules, and the metabolome. *Journal of Cheminformatics* **2020,** 12, (1), 43.
80. Durant, J. L.; Leland, B. A.; Henry, D. R.; Nourse, J. G., Reoptimization of MDL Keys for Use in Drug Discovery. *Journal of Chemical Information and Computer Sciences* **2002,** 42, (6), 1273-1280.
81. O'Boyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T.; Hutchison, G. R., Open Babel: An open chemical toolbox. *J. Cheminf.* **2011**.
82. Landrum, G.; Tosco, P.; Kelley, B.; sriniker; gedeck; NadineSchneider; Vianello, R.; Ric; Dalke, A.; Cole, B.; AlexanderSavelyev; Swain, M.; Turk, S.; N, D.; Vaucher, A.; Kawashima, E.; Wójcikowski, M.; Probst, D.; godin, g.; Cosgrove, D.; Pahl, A.; JP; Berenger, F.; strets123; JLVarjo; O'Boyle, N.; Fuller, P.; Jensen, J. H.; Sforna, G.; DoliathGavid *rdkit/rdkit: 2020_03_1 (Q1 2020) Release*, Release_2020_03_1; Zenodo: 2020.
83. Amigó, J. M.; Gálvez, J.; Villar, V. M., A review on molecular topology: applying graph theory to drug discovery and design. *Naturwissenschaften* **2009,** 96, (7), 749-761.
84. Ferreira, L. G.; Dos Santos, R. N.; Oliva, G.; Andricopulo, A. D., Molecular Docking and Structure-Based Drug Design Strategies. *Molecules* **2015,** 20, (7).
85. Ban, F.; Dalal, K.; Li, H.; LeBlanc, E.; Rennie, P. S.; Cherkasov, A., Best Practices of Computer-Aided Drug Discovery: Lessons Learned from the Development of a Preclinical Candidate for Prostate Cancer with a New Mechanism of Action. *Journal of Chemical Information and Modeling* **2017,** 57, (5), 1018-1028.
86. Stan, M., Discovery and design of nuclear fuels. *Materials Today* **2009,** 12, (11), 20-

doi:10.6342/NTU202403528

28.

87. Frenkel, D.; Smit, B., *Understanding Molecular Simulation: From Algorithms to Applications*. 2nd ed.; Academic Press: 2002; p 664.

88. Hossain, S.; Kabedev, A.; Parrow, A.; Bergström, C. A. S.; Larsson, P., Molecular simulation as a computational pharmaceutics tool to predict drug solubility, solubilization processes and partitioning. *European Journal of Pharmaceutics and Biopharmaceutics* **2019,** 137, 46-55.

89. Maginn, E. J.; Messerly, R. A.; Carlson, D. J.; Roe, D. R.; Elliott, J. R., Best Practices for Computing Transport Properties 1. Self-Diffusivity and Viscosity from Equilibrium Molecular Dynamics [Article v1.0]. *Living Journal of Computational Molecular Science* **2018,** 1, (1).

90. Orozco, G. A.; Moultos, O. A.; Jiang, H.; Economou, I. G.; Panagiotopoulos, A. Z., Molecular simulation of thermodynamic and transport properties for the H2O+NaCl system. *The Journal of Chemical Physics* **2014,** 141, (23), 234507.

91. Kataoka, Y.; Yamada, Y., Phase Diagram of a Lennard-Jones System by Molecular Dynamics Simulations. *Journal of Computer Chemistry, Japan* **2014,** 13, (2), 115-123.

92. Zhang, Y.; Maginn, E. J., A comparison of methods for melting point calculation using molecular dynamics simulations. *The Journal of Chemical Physics* **2012,** 136, (14), 144116.

93. Ytreberg, F. M.; Swendsen, R. H.; Zuckerman, D. M., Comparison of free energy methods for molecular systems. *The Journal of Chemical Physics* **2006,** 125, (18), 184114.

94. Joshi, S. Y.; Deshmukh, S. A., A review of advancements in coarse-grained molecular dynamics simulations. *Molecular Simulation* **2021,** 47, (10-11), 786-803.

95. Dubbeldam, D.; Walton, K. S.; Vlugt, T. J. H.; Calero, S., Design, Parameterization, and Implementation of Atomic Force Fields for Adsorption in Nanoporous Materials. *Advanced Theory and Simulations* **2019,** 2, (11), 1900135.

96. Sizova, O. V.; Skripnikov, L. V.; Sokolov, A. Y., Symmetry decomposition of quantum chemical bond orders. *Journal of Molecular Structure: THEOCHEM* **2008,** 870, (1), 1-9.

97. Parr, R. G.; Szentpály, L. v.; Liu, S., Electrophilicity Index. *Journal of the American Chemical Society* **1999,** 121, (9), 1922-1924.

98. Ochterski, J. W. *Thermochemistry in gaussian*; Gaussian Inc 2000.

doi:10.6342/NTU202403528

99. Curtiss, L. A.; Redfern, P. C.; Raghavachari, K., Gaussian-4 theory. *The Journal of Chemical Physics* **2007,** 126, (8), 084108.

100. Hsieh, C.-M.; Sandler, S. I.; Lin, S.-T., Improvements of COSMO-SAC for vapor–liquid and liquid–liquid equilibrium predictions. *Fluid Phase Equilibria* **2010,** 297, (1), 90-97.

101. Klamt, A., Conductor-like Screening Model for Real Solvents: A New Approach to the Quantitative Calculation of Solvation Phenomena. *The journal of Physical Chemistry* **1995,** 99, (7), 2224-2235.

102. Klamt, A.; Schüürmann, G., COSMO: a new approach to dielectric screening in solvents with explicit expressions for the screening energy and its gradient. *J. Chem. Soc., Perkin Trans. 2* **1993**, 799-805.

103. Tsai, C.-C.; Lin, S.-T., Integration of modern computational chemistry and ASPEN PLUS for chemical process design. *Aiche Journal* **2020,** 66, (10), e16987.

104. Tanaji, T. T.; Santosh, A. K.; Alan, C. R., Successful Applications of Computer Aided Drug Discovery: Moving Drugs from Concept to the Clinic. *Current Topics in Medicinal Chemistry* **2010,** 10, (1), 127-141.

105. Polishchuk, P., CReM: chemically reasonable mutations framework for structure generation. *Journal of Cheminformatics* **2020,** 12, (1), 28.

106. Hoksza, D.; Škoda, P.; Voršilák, M.; Svozil, D., Molpher: a software framework for systematic chemical space exploration. *Journal of Cheminformatics* **2014,** 6, (1), 7.

107. Hoksza, D.; Svozil, D. In *Exploration of Chemical Space by Molecular Morphing*, 13th IEEE International Conference on BioInformatics and BioEngineering, 24-26 October 2011, 2011; 2011; pp 201-208.

108. van Deursen, R.; Reymond, J.-L., Chemical Space Travel. *ChemMedChem* **2007,** 2, (5), 636-640.

109. Lameijer, E.-W.; Kok, J. N.; Bäck, T.; Ijzerman, A. P., The Molecule Evoluator. An Interactive Evolutionary Algorithm for the Design of Drug-Like Molecules. *Journal of Chemical Information and Modeling* **2006,** 46, (2), 545-552.

110. Jensen, J. H., A graph-based genetic algorithm and generative model/Monte Carlo tree search for the exploration of chemical space. *Chemical Science* **2019,** 10, (12), 3567-3572.

doi:10.6342/NTU202403528

111. Brown, N.; Fiscato, M.; Segler, M. H. S.; Vaucher, A. C., GuacaMol: Benchmarking Models for de Novo Molecular Design. *Journal of Chemical Information and Modeling* **2019,** 59, (3), 1096-1108.

112. Leguy, J.; Cauchy, T.; Glavatskikh, M.; Duval, B.; Da Mota, B., EvoMol: a flexible and interpretable evolutionary algorithm for unbiased de novo molecular generation. *Journal of Cheminformatics* **2020,** 12, (1), 55.

113. Kerstjens, A.; De Winter, H., LEADD: Lamarckian evolutionary algorithm for de novo drug design. *Journal of Cheminformatics* **2022,** 14, (1), 3.

114. Green, D. V. S.; Pickett, S.; Luscombe, C.; Senger, S.; Marcus, D.; Meslamani, J.; Brett, D.; Powell, A.; Masson, J., BRADSHAW: a system for automated molecular design. *Journal of Computer-Aided Molecular Design* **2020,** 34, (7), 747-765.

115. Kutchukian, P. S.; Lou, D.; Shakhnovich, E. I., FOG: Fragment Optimized Growth Algorithm for the de Novo Generation of Molecules Occupying Druglike Chemical Space. *Journal of Chemical Information and Modeling* **2009,** 49, (7), 1630-1642.

116. Yuan, Y.; Pei, J.; Lai, L., LigBuilder V3: A Multi-Target de novo Drug Design Approach. *Frontiers in Chemistry* **2020,** 8.

117. Yuan, Y.; Pei, J.; Lai, L., LigBuilder 2: A Practical de Novo Drug Design Approach. *Journal of Chemical Information and Modeling* **2011,** 51, (5), 1083-1091.

118. Firth, N. C.; Atrash, B.; Brown, N.; Blagg, J., MOARF, an Integrated Workflow for Multiobjective Optimization: Implementation, Synthesis, and Biological Evaluation. *Journal of Chemical Information and Modeling* **2015,** 55, (6), 1169-1180.

119. Huang, Q.; Li, L.-L.; Yang, S.-Y., PhDD: A new pharmacophore-based de novo design method of drug-like molecules combined with assessment of synthetic accessibility. *Journal of Molecular Graphics and Modelling* **2010,** 28, (8), 775-787.

120. Durrant, J. D.; Lindert, S.; McCammon, J. A., AutoGrow 3.0: An improved algorithm for chemically tractable, semi-automated protein inhibitor design. *Journal of Molecular Graphics and Modelling* **2013,** 44, 104-112.

121. Durrant, J. D.; Amaro, R. E.; McCammon, J. A., AutoGrow: A Novel Algorithm for Protein Inhibitor Design. *Chemical Biology & Drug Design* **2009,** 73, (2), 168-178.

122. Lowe, D., Chemical reactions from US patents (1976-Sep2016). In figshare: 2017.

123. Batiste, L.; Unzue, A.; Dolbois, A.; Hassler, F.; Wang, X.; Deerain, N.; Zhu, J.;

doi:10.6342/NTU202403528

Spiliotopoulos, D.; Nevado, C.; Caflisch, A., Chemical Space Expansion of Bromodomain Ligands Guided by in Silico Virtual Couplings (AutoCouple). *ACS Central Science* **2018,** 4, (2), 180-188.

124. Dalke, A.; Hert, J.; Kramer, C., mmpdb: An Open-Source Matched Molecular Pair Platform for Large Multiproperty Data Sets. *Journal of Chemical Information and Modeling* **2018,** 58, (5), 902-910.

125. Genheden, S.; Thakkar, A.; Chadimová, V.; Reymond, J.-L.; Engkvist, O.; Bjerrum, E., AiZynthFinder: a fast, robust and flexible open-source software for retrosynthetic planning. *Journal of Cheminformatics* **2020,** 12, (1), 70.

126. Coley, C. W.; Green, W. H.; Jensen, K. F., Machine Learning in Computer-Aided Synthesis Planning. *Accounts of Chemical Research* **2018,** 51, (5), 1281-1289.

127. Coley, C. W.; Green, W. H.; Jensen, K. F., RDChiral: An RDKit Wrapper for Handling Stereochemistry in Retrosynthetic Template Extraction and Application. *Journal of Chemical Information and Modeling* **2019,** 59, (6), 2529-2537.

128. Hartenfeller, M.; Zettl, H.; Walter, M.; Rupp, M.; Reisen, F.; Proschak, E.; Weggen, S.; Stark, H.; Schneider, G., DOGS: Reaction-Driven de novo Design of Bioactive Compounds. *PLoS Computational Biology* **2012,** 8, (2), e1002380.

129. Merk, D.; Grisoni, F.; Friedrich, L.; Gelzinyte, E.; Schneider, G., Computer-Assisted Discovery of Retinoid X Receptor Modulating Natural Products and Isofunctional Mimetics. *Journal of Medicinal Chemistry* **2018,** 61, (12), 5442-5447.

130. Beccari, A. R.; Cavazzoni, C.; Beato, C.; Costantino, G., LiGen: A High Performance Workflow for Chemistry Driven de Novo Design. *Journal of Chemical Information and Modeling* **2013,** 53, (6), 1518-1527.

131. Vinkers, H. M.; de Jonge, M. R.; Daeyaert, F. F. D.; Heeres, J.; Koymans, L. M. H.; van Lenthe, J. H.; Lewi, P. J.; Timmerman, H.; Van Aken, K.; Janssen, P. A. J., SYNOPSIS: SYNthesize and OPtimize System in Silico. *Journal of Medicinal Chemistry* **2003,** 46, (13), 2765-2773.

132. Zahra, B.; Siti Mariyam Hj, S., A Review of Population-based Meta-Heuristic Algorithm. *International Journal of Advances in Soft Computing & Its Applications* **2013,** 5, (1), 1-35.

133. Clark, D. E.; Westhead, D. R., Evolutionary algorithms in computer-aided

doi:10.6342/NTU202403528

molecular design. *J. Comput.-Aided Mol. Des.* **1996,** 10, (4), 337-358.

134. Costa, L.; Oliveira, P., Evolutionary algorithms approach to the solution of mixed integer non-linear programming problems. *Computers and Chemical Engineering* **2001,** 25, 257-266.

135. Devillers, J., *Genetic Algorithms in Molecular Modeling*. Elsevier Science & Technology Books: 1996.

136. Goldberg, D. E., *Genetic Algorithms in Search, Optimization, and Machine Learning*. 1st ed.; Addison-Wesley Longman Publishing Co.: 1989.

137. Androulakis, I. P.; Venkatasubramanian, V., A genetic algorithmic framework for process design and optimization. *Comput. Chem. Eng.* **1991,** 15, (4), 217-228.

138. Wright, A. H., Genetic Algorithms for Real Parameter Optimization. *Foundations of Genetic Algorithms* **1999**.

139. Ourique, J. E.; Silva Telles, A., Computer-aided molecular design with simulated annealing and molecular graphs. *Computers & Chemical Engineering* **1998,** 22, S615-S618.

140. Aarts, E. H. L.; Laarhoven, P. J. M. v., Statistical cooling : a general approach to combinatorial optimization problems. *Philips Journal of Research* **1985,** 40, (4), 193-226.

141. Harvey, I., Species Adaptation Genetic Algorithms: A Basis for a Continuing SAGA. *Proceedings of the First European Conference on Artificial Life: Toward a Practice of Autonomous Systems* **1992**.

142. Zhang, J.; Qin, L.; Peng, D.; Zhou, T.; Cheng, H.; Chen, L.; Qi, Z., COSMO-descriptor based computer-aided ionic liquid design for separation processes: Part II: Task-specific design for extraction processes. *Chemical Engineering Science* **2017,** 162, 364-374.

143. Zhang, J.; Peng, D.; Song, Z.; Zhou, T.; Cheng, H.; Chen, L.; Qi, Z., COSMO-descriptor based computer-aided ionic liquid design for separation processes. Part I: Modified group contribution methodology for predicting surface charge density profile of ionic liquids. *Chemical Engineering Science* **2017,** 162, 355-363.

144. Liu, B.; Wen, Y.; Zhang, X., Development of CAMD based on the hybrid gene algorithm and simulated annealing algorithm and the application on solvent selection. *The Canadian Journal of Chemical Engineering* **2017,** 95, (4), 767-774.

145. Diwekar, U. M.; Gebreslassie, B. H., Efficient ant colony optimization (EACO)

doi:10.6342/NTU202403528

algorithm for deterministic optimization. *International Journal of Swarm Intelligence and Evolutionary Computation* **2016,** 5, (131).

146. Dorigo, M.; Caro, G. D.; Gambardella, L. M., Ant Algorithms for Discrete Optimization. *Artificial Life* **1999,** 5, (2), 137-172.

147. Gebreslassie, B. H.; Diwekar, U. M., Efficient ant colony optimization for computer aided molecular design: Case study solvent selection problem. *Computers & Chemical Engineering* **2015,** 78, 1-9.

148. Pham, D. T.; Karaboga, D., *Intelligent Optimisation Techniques - Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks*. 1st ed.; Springer, London: 2000.

149. Pham, D. T.; Karaboga, D., Tabu Search. In *Intelligent Optimisation Techniques: Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks*, Pham, D. T.; Karaboga, D., Eds. Springer London: London, 2000; pp 149-186.

150. Glover, F.; Taillard, E.; Taillard, E., A user's guide to tabu search. *Annals of Operations Research* **1993,** 41, (1), 1-28.

151. Browne, C. B.; Powley, E.; Whitehouse, D.; Lucas, S. M.; Cowling, P. I.; Rohlfshagen, P.; Tavener, S.; Perez, D.; Samothrakis, S.; Colton, S., A Survey of Monte Carlo Tree Search Methods. *IEEE Transactions on Computational Intelligence and AI in Games* **2012,** 4, (1), 1-43.

152. M. Dieb, T.; Ju, S.; Yoshizoe, K.; Hou, Z.; Shiomi, J.; Tsuda, K., MDTS: automatic complex materials design using Monte Carlo tree search. *Science and Technology of Advanced Materials* **2017,** 18, (1), 498-503.

153. Ishida, S.; Aasawat, T.; Sumita, M.; Katouda, M.; Yoshizawa, T.; Yoshizoe, K.; Tsuda, K.; Terayama, K., ChemTSv2: Functional molecular design using de novo molecule generator. *WIREs Computational Molecular Science* **2023,** n/a, (n/a), e1680.

154. Yang, X.; Zhang, J.; Yoshizoe, K.; Terayama, K.; Tsuda, K., ChemTS: an efficient python library for de novo molecular generation. *Science and Technology of Advanced Materials* **2017,** 18, (1), 972-976.

155. You, F. Cornell University Computational Optimization Open Textbook - Optimization Wiki. https://optimization.cbe.cornell.edu/index.php?title=Main_Page (2023/6/1),

doi:10.6342/NTU202403528

156. Ryoo, H. S.; Sahinidis, N. V., A branch-and-reduce approach to global optimization. *Journal of Global Optimization* **1996,** 8, (2), 107-138.

157. Viswanathan, J.; Grossmann, I. E., A Combined Penalty Function and Outer-Approaximation Method for MINLP Optimization. *Comput. Chem. Eng.* **1990,** 14, (7), 769-782.

158. Horst, R., Deterministic methods in constrained global optimization: Some recent advances and new fields of application. *Naval Research Logistics (NRL)* **1990,** 37, (4), 433-471.

159. Gopinath, S.; Jackson, G.; Galindo, A.; Adjiman, C. S., Outer approximation algorithm with physical domain reduction for computer-aided molecular and separation process design. *Aiche Journal* **2016,** 62, (9), 3484-3504.

160. Viswanathan, J.; Grossmann, I. E., A combined penalty function and outer-approximation method for MINLP optimization. *Computers & Chemical Engineering* **1990,** 14, (7), 769-782.

161. Harper, P. M.; Gani, R.; Kolar, P.; Ishikawa, T., Computer-aided molecular design with combined molecular modeling and group contribution. *Fluid Phase Equilibria* **1999,** 158-160, 337-347.

162. Hsu, H.-H.; Huang, C.-H.; Lin, S.-T., New Data Structure for Computational Molecular Design with Atomic or Fragment Resolution. *Journal of Chemical Information and Modeling* **2019,** 59, (9), 3703-3713.

163. Huang, C.-H.; Lin, S.-T., MARS Plus: An Improved Molecular Design Tool for Complex Compounds Involving Ionic, Stereo, and Cis–Trans Isomeric Structures. *Journal of Chemical Information and Modeling* **2023,** 63, (24), 7711-7728.

164. Frisch, M. J.; Trucks, G. W.; Schlegel, H. B.; Scuseria, G. E.; Robb, M. A.; Cheeseman, J. R.; Scalmani, G.; Barone, V.; Petersson, G. A.; Nakatsuji, H.; Li, X.; Caricato, M.; Marenich, A. V.; Bloino, J.; Janesko, B. G.; Gomperts, R.; Mennucci, B.; Hratchian, H. P.; Ortiz, J. V.; Izmaylov, A. F.; Sonnenberg, J. L.; Williams; Ding, F.; Lipparini, F.; Egidi, F.; Goings, J.; Peng, B.; Petrone, A.; Henderson, T.; Ranasinghe, D.; Zakrzewski, V. G.; Gao, J.; Rega, N.; Zheng, G.; Liang, W.; Hada, M.; Ehara, M.; Toyota, K.; Fukuda, R.; Hasegawa, J.; Ishida, M.; Nakajima, T.; Honda, Y.; Kitao, O.; Nakai, H.; Vreven, T.; Throssell, K.; Montgomery Jr., J. A.; Peralta, J. E.; Ogliaro, F.; Bearpark, M. J.; Heyd, J. J.; Brothers, E. N.; Kudin, K. N.;

doi:10.6342/NTU202403528

Staroverov, V. N.; Keith, T. A.; Kobayashi, R.; Normand, J.; Raghavachari, K.; Rendell, A. P.; Burant, J. C.; Iyengar, S. S.; Tomasi, J.; Cossi, M.; Millam, J. M.; Klene, M.; Adamo, C.; Cammi, R.; Ochterski, J. W.; Martin, R. L.; Morokuma, K.; Farkas, O.; Foresman, J. B.; Fox, D. J. *Gaussian 16 Rev. A.01*, Wallingford, CT, 2016.

165. Bredas, J.-L., Mind the gap! *Materials Horizons* **2014,** 1, (1), 17-19.

166. Parr, R. G.; Donnelly, R. A.; Levy, M.; Palke, W. E., Electronegativity: The density functional viewpoint. *The Journal of Chemical Physics* **1978,** 68, (8), 3801-3807.

167. Chattaraj, P. K.; Giri, S.; Duley, S., Update 2 of: Electrophilicity Index. *Chemical Reviews* **2011,** 111, (2), PR43-PR75.

168. Ertl, P.; Schuffenhauer, A., Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of Cheminformatics* **2009,** 1, (1), 8.

169. Coley, C. W.; Rogers, L.; Green, W. H.; Jensen, K. F., SCScore: Synthetic Complexity Learned from a Reaction Corpus. *Journal of Chemical Information and Modeling* **2018,** 58, (2), 252-261.

170. Ertl, P.; Roggo, S.; Schuffenhauer, A., Natural Product-likeness Score and Its Application for Prioritization of Compound Libraries. *Journal of Chemical Information and Modeling* **2008,** 48, (1), 68-74.

171. Getting Started with the RDKit in Python: List of Available Descriptors. https://www.rdkit.org/docs/GettingStartedInPython.html#list-of-available-descriptors (2024/06/17),

172. Bell, I. H.; Mickoleit, E.; Hsieh, C.-M.; Lin, S.-T.; Vrabec, J.; Breitkopf, C.; Jäger, A., A Benchmark Open-Source Implementation of COSMO-SAC. *Journal of Chemical Theory and Computation* **2020,** 16, (4), 2635-2646.

173. Lin, S.-T.; Hsieh, C.-M.; Lee, M.-T., Solvation and chemical engineering thermodynamics. *Journal of the Chinese Institute of Chemical Engineers* **2007,** 38, (5), 467-476.

174. Ben-Naim, A., *Solvation Thermodynamics*. first ed. ed.; Plenum Press: New York, 1987.

175. Lin, S.-T.; Sandler, S. I., A Priori Phase Equilibrium Prediction from a Segment Contribution Solvation Model. *Industrial & Engineering Chemistry Research* **2002,** 41, (5),

doi:10.6342/NTU202403528

899-913.

176. Hsieh, C.-M.; Lin, S.-T., Determination of cubic equation of state parameters for pure fluids from first principle solvation calculations. *Aiche Journal* **2008,** 54, (8), 2174-2181.

177. Hsieh, C. M.; Lin, S. T., Determination of cubic equation of state parameters for pure fluids from first principle solvation calculations. *AIChE J.* **2008,** 54, (8), 2174-2181.

178. Hsieh, C.-M.; Lin, S.-T., First-Principles Predictions of Vapor−Liquid Equilibria for Pure and Mixture Fluids from the Combined Use of Cubic Equations of State and Solvation Calculations. *Industrial & Engineering Chemistry Research* **2009,** 48, (6), 3197-3205.

179. Hsieh, C.-M.; Lin, S.-T.; Vrabec, J., Considering the dispersive interactions in the COSMO-SAC model for more accurate predictions of fluid phase behavior. *Fluid Phase Equilibria* **2014,** 367, 109-116.

180. Lin, S.-T.; Sandler, S. I., Infinite dilution activity coefficients from ab initio solvation calculations. *Aiche Journal* **1999,** 45, (12), 2606-2618.

181. Staverman, A. J., The Entropy of High Polymer Solutions. *Recueil des Travaus Chimiques des Pays-Bas* **1950,** 69, 163-174.

182. Guggenheim, E. A., *Mixtures: The theory of the equilibrium properties of some simple classes of mixtures, solutions and alloys.* Clarendon Press: Oxford, 1952.

183. Parr, R. G.; Weitao, Y., *Density-Functional Theory of Atoms and Molecules*. Oxford University Press: 1994.

184. Gelfand, I. M.; Fomin, S. V., *Calculus of Variations*. Dover Publications: 2012.

185. Mulliken, R. S., Electronic Structures of Molecules XI. Electroaffinity, Molecular Orbitals and Dipole Moments. *The Journal of Chemical Physics* **1935,** 3, (9), 573-585.

186. Jenkins, A. D., Interpretation of reactivity in radical polymerization—Radicals, monomers, and transfer agents: Beyond the Q-e scheme. *Journal of Polymer Science Part A: Polymer Chemistry* **1999,** 37, (2), 113-126.

187. Rogers, S. C.; Mackrodt, W. C.; Davis, T. P., Ab initio molecular orbital calculations on the Q-e scheme for predicting reactivity in free-radical copolymerization. *Polymer* **1994,** 35, (6), 1258-1267.

188. Chattaraj, P. K.; Giri, S.; Duley, S., Electrophilicity Equalization Principle. *The*

doi:10.6342/NTU202403528

Journal of Physical Chemistry Letters **2010,** 1, (7), 1064-1067.

189. Chattaraj, P. K.; Lee, H.; Parr, R. G., HSAB principle. *Journal of the American Chemical Society* **1991,** 113, (5), 1855-1856.

190. Mattson, C. A.; Messac, A., Pareto Frontier Based Concept Selection Under Uncertainty, with Visualization. *Optimization and Engineering* **2005,** 6, (1), 85-115.

191. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T., A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* **2002,** 6, (2), 182-197.

192. May, J. W.; Steinbeck, C., Efficient ring perception for the Chemistry Development Kit. *Journal of Cheminformatics* **2014,** 6, (1), 3.

193. Plotkin, M., Mathematical Basis of Ring-Finding Algorithms in CIDS. *Journal of Chemical Documentation* **1971,** 11, (1), 60-63.

194. Lee, C. J.; Kang, Y.-M.; Cho, K.-H.; No, K. T., A robust method for searching the smallest set of smallest rings with a path-included distance matrix. *Proceedings of the National Academy of Sciences* **2009,** 106, (41), 17355-17358.

195. Apodaca, R. L., A Smallest Set of Smallest Rings. In Metamolecular, LLC: 2020; Vol. 2021.

196. Nachbar, R. B., Molecular Evolution: Automated Manipulation of Hierarchical Chemical Topology and Its Application to Average Molecular Structures. *Genetic Programming and Evolvable Machines* **2000,** 1, (1), 57-94.

197. Yeung; Hong, S.; Corey, E. J., A Short Enantioselective Pathway for the Synthesis of the Anti-Influenza Neuramidase Inhibitor Oseltamivir from 1,3-Butadiene and Acrylic Acid. *Journal of the American Chemical Society* **2006,** 128, (19), 6310-6311.

198. Laborda, P.; Wang, S.-Y.; Voglmeir, J., Influenza Neuraminidase Inhibitors: Synthetic Approaches, Derivatives and Biological Activity. In *Molecules*, 2016; Vol. 21.

199. IPCC, *Global Warming of 1.5°C: An IPCC Special Report on the impacts of global warming of 1.5°C above pre-industrial levels and related global greenhouse gas emission pathways, in the context of strengthening the global response to the threat of climate change, sustainable development, and efforts to eradicate poverty*. Cambridge University Press: Cambridgee, UK and New York, NY, USA., 2018.

200. Friedlingstein, P.; O'Sullivan, M.; Jones, M. W.; Andrew, R. M.; Gregor, L.; Hauck,

doi:10.6342/NTU202403528

J.; Le Quéré, C.; Luijkx, I. T.; Olsen, A.; Peters, G. P.; Peters, W.; Pongratz, J.; Schwingshackl, C.; Sitch, S.; Canadell, J. G.; Ciais, P.; Jackson, R. B.; Alin, S. R.; Alkama, R.; Arneth, A.; Arora, V. K.; Bates, N. R.; Becker, M.; Bellouin, N.; Bittig, H. C.; Bopp, L.; Chevallier, F.; Chini, L. P.; Cronin, M.; Evans, W.; Falk, S.; Feely, R. A.; Gasser, T.; Gehlen, M.; Gkritzalis, T.; Gloege, L.; Grassi, G.; Gruber, N.; Gürses, Ö.; Harris, I.; Hefner, M.; Houghton, R. A.; Hurtt, G. C.; Iida, Y.; Ilyina, T.; Jain, A. K.; Jersild, A.; Kadono, K.; Kato, E.; Kennedy, D.; Klein Goldewijk, K.; Knauer, J.; Korsbakken, J. I.; Landschützer, P.; Lefèvre, N.; Lindsay, K.; Liu, J.; Liu, Z.; Marland, G.; Mayot, N.; McGrath, M. J.; Metzl, N.; Monacci, N. M.; Munro, D. R.; Nakaoka, S. I.; Niwa, Y.; O'Brien, K.; Ono, T.; Palmer, P. I.; Pan, N.; Pierrot, D.; Pocock, K.; Poulter, B.; Resplandy, L.; Robertson, E.; Rödenbeck, C.; Rodriguez, C.; Rosan, T. M.; Schwinger, J.; Séférian, R.; Shutler, J. D.; Skjelvan, I.; Steinhoff, T.; Sun, Q.; Sutton, A. J.; Sweeney, C.; Takao, S.; Tanhua, T.; Tans, P. P.; Tian, X.; Tian, H.; Tilbrook, B.; Tsujino, H.; Tubiello, F.; van der Werf, G. R.; Walker, A. P.; Wanninkhof, R.; Whitehead, C.; Willstrand Wranne, A.; Wright, R.; Yuan, W.; Yue, C.; Yue, X.; Zaehle, S.; Zeng, J.; Zheng, B., Global Carbon Budget 2022. *Earth Syst. Sci. Data* **2022,** 14, (11), 4811-4900.

201. Liu, Z.; Deng, Z.; Davis, S.; Ciais, P., Monitoring global carbon emissions in 2022. *Nature Reviews Earth & Environment* **2023,** 4, (4), 205-206.

202. IEA *CO2 Emissions in 2022*; International Energy Agency (IEA): Paris, 2023.

203. IEAGHG *Assessment of emerging CO2 capture technologies and their potential to reduce cost*; International Energy Agency (IEA): 2014.

204. Sifat, S. N.; Haseli, Y., A Critical Review of CO2 Capture Technologies and Prospects for Clean Power Generation. *Energies* **2019,** 12, (21).

205. Leung, D. Y. C.; Caramanna, G.; Maroto-Valer, M. M., An overview of current status of carbon dioxide capture and storage technologies. *Renewable and Sustainable Energy Reviews* **2014,** 39, 426-443.

206. Aaron, D.; Tsouris, C., Separation of CO2 from Flue Gas: A Review. *Separation Science and Technology* **2005,** 40, (1-3), 321-348.

207. Ramdin, M.; de Loos, T. W.; Vlugt, T. J. H., State-of-the-Art of CO2 Capture with Ionic Liquids. *Industrial & Engineering Chemistry Research* **2012,** 51, (24), 8149-8177.

208. Boot-Handford, M. E.; Abanades, J. C.; Anthony, E. J.; Blunt, M. J.; Brandani, S.; Mac Dowell, N.; Fernández, J. R.; Ferrari, M.-C.; Gross, R.; Hallett, J. P.; Haszeldine, R. S.;

doi:10.6342/NTU202403528

Heptonstall, P.; Lyngfelt, A.; Makuch, Z.; Mangano, E.; Porter, R. T. J.; Pourkashanian, M.; Rochelle, G. T.; Shah, N.; Yao, J. G.; Fennell, P. S., Carbon capture and storage update. *Energy & Environmental Science* **2014,** 7, (1), 130-189.

209. Yu, C.-H.; Huang, C.-H.; Tan, C.-S., A Review of CO2 Capture by Absorption and Adsorption. *Aerosol and Air Quality Research* **2012,** 12, (5), 745-769.

210. Kapetaki, Z.; Brandani, P.; Brandani, S.; Ahn, H., Process simulation of a dual-stage Selexol process for 95% carbon capture efficiency at an integrated gasification combined cycle power plant. *International Journal of Greenhouse Gas Control* **2015,** 39, 17-26.

211. Zhang, X.; Song, Z.; Gani, R.; Zhou, T., Comparative Economic Analysis of Physical, Chemical, and Hybrid Absorption Processes for Carbon Capture. *Industrial & Engineering Chemistry Research* **2020,** 59, (5), 2005-2012.

212. *Developing a Pipeline Infrastructure for CO2 Capture and Storage: Issues and Challenges*; ICF International: 2/1, 2009.

213. Li, T.; Yang, C.; Tantikhajorngosol, P.; Sema, T.; Tontiwachwuthikul, P., Studies on advanced configurations of post-combustion CO2 capture process applied to cement plant flue gases. *Carbon Capture Science & Technology* **2022,** 4, 100064.

214. Laribi, S.; Dubois, L.; De Weireld, G.; Thomas, D., Study of the post-combustion CO2 capture process by absorption-regeneration using amine solvents applied to cement plant flue gases with high CO2 contents. *International Journal of Greenhouse Gas Control* **2019,** 90, 102799.

215. Aouini, I.; Ledoux, A.; Estel, L.; Mary, S., Pilot Plant Studies for CO2 Capture from Waste Incinerator Flue Gas Using MEA Based Solvent. *Oil Gas Sci. Technol. – Rev. IFP Energies nouvelles* **2014,** 69, (6), 1091-1104.

216. Padurean, A.; Cormos, C.-C.; Agachi, P.-S., Pre-combustion carbon dioxide capture by gas–liquid absorption for Integrated Gasification Combined Cycle power plants. *International Journal of Greenhouse Gas Control* **2012,** 7, 1-11.

217. Alptekin, G. O.; Jayaraman, A.; Bonnema, M.; Gribble, D. *Integrated Water-Gas-Shift Pre-combustion Carbon Capture Process*; United States, 2022-02-04, 2022.

218. Nazir, S. M.; Bolland, O.; Amini, S., Analysis of Combined Cycle Power Plants with Chemical Looping Reforming of Natural Gas and Pre-Combustion CO2 Capture. In *Energies*, 2018; Vol. 11.

doi:10.6342/NTU202403528

219. Voldsund, M.; Jordal, K.; Anantharaman, R., Hydrogen production with CO2 capture. *International Journal of Hydrogen Energy* **2016,** 41, (9), 4969-4992.

220. Berstad, D.; Anantharaman, R.; Nekså, P., Low-temperature CCS from an IGCC Power Plant and Comparison with Physical Solvents. *Energy Procedia* **2013,** 37, 2204-2211.

221. Romano, M. C.; Chiesa, P.; Lozza, G., Pre-combustion CO2 capture from natural gas power plants, with ATR and MDEA processes. *International Journal of Greenhouse Gas Control* **2010,** 4, (5), 785-797.

222. Porter, R. T. J.; Fairweather, M.; Pourkashanian, M.; Woolley, R. M., The range and level of impurities in CO2 streams from different carbon capture sources. *International Journal of Greenhouse Gas Control* **2015,** 36, 161-174.

223. Anheden, M.; Burchhardt, U.; Ecke, H.; Faber, R.; Jidinger, O.; Giering, R.; Kass, H.; Lysk, S.; Ramström, E.; Yan, J., Overview of operational experience and results from test activities in Vattenfall's 30 MWth oxyfuel pilot plant in Schwarze Pumpe. *Energy Procedia* **2011,** 4, 941-950.

224. Zheng, L., *Oxy-fuel combustion for power generation and carbon dioxide (CO2) capture*. Elsevier: 2011.

225. Han, K.; Ahn, C. K.; Lee, M. S.; Rhee, C. H.; Kim, J. Y.; Chun, H. D., Current status and challenges of the ammonia-based CO2 capture technologies toward commercialization. *International Journal of Greenhouse Gas Control* **2013,** 14, 270-281.

226. Singh, A.; Stéphenne, K., Shell Cansolv CO2 capture technology: Achievement from First Commercial Plant. *Energy Procedia* **2014,** 63, 1678-1685.

227. Chong, F. K.; Foo, D. C. Y.; Eljack, F. T.; Atilhan, M.; Chemmangattuvalappil, N. G., Ionic liquid design for enhanced carbon dioxide capture by computer-aided molecular design approach. *Clean Technologies and Environmental Policy* **2015,** 17, (5), 1301-1312.

228. Oko, E.; Wang, M.; Joel, A. S., Current status and future development of solvent-based carbon capture. *International Journal of Coal Science & Technology* **2017,** 4, (1), 5-14.

229. Ahn, H.; Luberti, M.; Liu, Z.; Brandani, S., Process configuration studies of the amine capture process for coal-fired power plants. *International Journal of Greenhouse Gas Control* **2013,** 16, 29-40.

doi:10.6342/NTU202403528

230. Brennecke, J. F.; Gurkan, B. E., Ionic Liquids for CO2 Capture and Emission Reduction. *The Journal of Physical Chemistry Letters* **2010,** 1, (24), 3459-3464.

231. Aghaie, M.; Rezaei, N.; Zendehboudi, S., A systematic review on CO2 capture with ionic liquids: Current status and future prospects. *Renewable and Sustainable Energy Reviews* **2018,** 96, 502-525.

232. Jiang, W.; Li, X.; Gao, G.; Wu, F.; Luo, C.; Zhang, L., Advances in applications of ionic liquids for phase change CO2 capture. *Chemical Engineering Journal* **2022,** 445, 136767.

233. Zeng, S.; Zhang, X.; Bai, L.; Zhang, X.; Wang, H.; Wang, J.; Bao, D.; Li, M.; Liu, X.; Zhang, S., Ionic-Liquid-Based CO2 Capture Systems: Structure, Interaction and Process. *Chemical Reviews* **2017,** 117, (14), 9625-9673.

234. Sharma, P.; Choi, S.-H.; Park, S.-D.; Baek, I.-H.; Lee, G.-S., Selective chemical separation of carbondioxide by ether functionalized imidazolium cation based ionic liquids. *Chemical Engineering Journal* **2012,** 181-182, 834-841.

235. Cui, G.; Wang, J.; Zhang, S., Active chemisorption sites in functionalized ionic liquids for carbon capture. *Chemical Society Reviews* **2016,** 45, (15), 4307-4339.

236. Liu, Y.; Dai, Z.; Zhang, Z.; Zeng, S.; Li, F.; Zhang, X.; Nie, Y.; Zhang, L.; Zhang, S.; Ji, X., Ionic liquids/deep eutectic solvents for CO2 capture: Reviewing and evaluating. *Green Energy & Environment* **2021,** 6, (3), 314-328.

237. Izgorodina, E. I.; Hodgson, J. L.; Weis, D. C.; Pas, S. J.; MacFarlane, D. R., Physical Absorption Of CO2 in Protic and Aprotic Ionic Liquids: An Interaction Perspective. *The Journal of Physical Chemistry B* **2015,** 119, (35), 11748-11759.

238. Shannon, M. S.; Tedstone, J. M.; Danielsen, S. P. O.; Hindman, M. S.; Irvin, A. C.; Bara, J. E., Free Volume as the Basis of Gas Solubility and Selectivity in Imidazolium-Based Ionic Liquids. *Industrial & Engineering Chemistry Research* **2012,** 51, (15), 5565-5576.

239. Rao, S. S.; Gejji, S. P., CO2 Absorption Using Fluorine Functionalized Ionic Liquids: Interplay of Hydrogen and σ-Hole Interactions. *The Journal of Physical Chemistry A* **2016,** 120, (8), 1243-1260.

240. Lin, H.; Freeman, B. D., Materials selection guidelines for membranes that remove CO2 from gas mixtures. *Journal of Molecular Structure* **2005,** 739, (1), 57-74.

241. Davies, J. A.; Griffiths, P. C., A Phenomenological Approach to Separating the

doi:10.6342/NTU202403528

Effects of Obstruction and Binding for the Diffusion of Small Molecules in Polymer Solutions. *Macromolecules* **2003,** 36, (3), 950-952.

242. Rogers, R. D.; Seddon, K. R., Ionic Liquids - Solvents of the Future? *Science* **2003,** 302, (5646), 792.

243. Seddon, K. R., Ionic Liquids for Clean Technology. *Journal of Chemical Technology & Biotechnology* **1997,** 68, (4), 351-356.

244. Ghandi, K., A Review of Ionic Liquids, Their Limits and Applications. *Green and Sustainable Chemistry* **2014,** Vol.04No.01, 10.

245. Marsh, K. N.; Boxall, J. A.; Lichtenthaler, R., Room temperature ionic liquids and their mixtures—a review. *Fluid Phase Equilibria* **2004,** 219, (1), 93-98.

246. Zhang, J.; Qiao, Y.; Wang, W.; Misch, R.; Hussain, K.; Agar, D. W., Development of an Energy-efficient CO2 Capture Process Using Thermomorphic Biphasic Solvents. *Energy Procedia* **2013,** 37, 1254-1261.

247. Mota-Martinez, M. T.; Brandl, P.; Hallett, J. P.; Mac Dowell, N., Challenges and opportunities for the utilisation of ionic liquids as solvents for CO2 capture. *Molecular Systems Design & Engineering* **2018,** 3, (3), 560-571.

248. Wang, C.; Luo, X.; Zhu, X.; Cui, G.; Jiang, D.-e.; Deng, D.; Li, H.; Dai, S., The strategies for improving carbon dioxide chemisorption by functionalized ionic liquids. *RSC Advances* **2013,** 3, (36), 15518-15527.

249. Farahipour, R.; Mehrkesh, A.; Karunanithi, A. T., A systematic screening methodology towards exploration of ionic liquids for CO2 capture processes. *Chemical Engineering Science* **2016,** 145, 126-132.

250. Wang, J.; Song, Z.; Cheng, H.; Chen, L.; Deng, L.; Qi, Z., Multilevel screening of ionic liquid absorbents for simultaneous removal of CO2 and H2S from natural gas. *Separation and Purification Technology* **2020,** 248, 117053.

251. Mukhopadhyay, M., A thermodynamic method based upon the theory of regular solutions for selection of solvents and process conditions for aromatics extraction. *Journal of Chemical Technology and Biotechnology* **1979,** 29, (10), 634-641.

252. Hospital-Benito, D.; Lemus, J.; Moya, C.; Santiago, R.; Palomar, J., Process analysis overview of ionic liquids on CO2 chemical capture. *Chemical Engineering Journal* **2020,** 390, 124509.

doi:10.6342/NTU202403528

253. Shama, V. M.; Swami, A. R.; Aniruddha, R.; Sreedhar, I.; Reddy, B. M., Process and engineering aspects of carbon capture by ionic liquids. *Journal of CO2 Utilization* **2021,** 48, 101507.

254. Anthony, J. L.; Anderson, J. L.; Maginn, E. J.; Brennecke, J. F., Anion Effects on Gas Solubility in Ionic Liquids. *The Journal of Physical Chemistry B* **2005,** 109, (13), 6366-6374.

255. Lee, B.-S.; Lin, S.-T., Screening of ionic liquids for CO2 capture using the COSMO-SAC model. *Chemical Engineering Science* **2015,** 121, 157-168.

256. Dong, Q.; Muzny, C. D.; Kazakov, A.; Diky, V.; Magee, J. W.; Widegren, J. A.; Chirico, R. D.; Marsh, K. N.; Frenkel, M., ILThermo: A Free-Access Web Database for Thermodynamic Properties of Ionic Liquids. *Journal of Chemical & Engineering Data* **2007,** 52, (4), 1151-1159.

257. Kazakov, A.; Magee, J. W.; Chirico, R. D.; Paulechka, E.; Diky, V.; Muzny, C. D.; Kroenlein, K.; Frenkel, M. NIST Standard Reference Database 147: NIST Ionic Liquids Database - (ILThermo), Version 2.0. http://ilthermo.boulder.nist.gov

258. Yokozeki, A.; Shiflett, M. B.; Junk, C. P.; Grieco, L. M.; Foo, T., Physical and Chemical Absorptions of Carbon Dioxide in Room-Temperature Ionic Liquids. *The Journal of Physical Chemistry B* **2008,** 112, (51), 16654-16663.

259. Kurnia, K. A.; Harris, F.; Wilfred, C. D.; Abdul Mutalib, M. I.; Murugesan, T., Thermodynamic properties of CO2 absorption in hydroxyl ammonium ionic liquids at pressures of (100–1600)kPa. *The Journal of Chemical Thermodynamics* **2009,** 41, (10), 1069-1073.

260. Gupta, M.; da Silva, E. F.; Hartono, A.; Svendsen, H. F., Theoretical Study of Differential Enthalpy of Absorption of CO2 with MEA and MDEA as a Function of Temperature. *The Journal of Physical Chemistry B* **2013,** 117, (32), 9457-9468.

261. Weininger, D., SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences* **1988,** 28, (1), 31-36.

262. Frisch, M. J.; Trucks, G. W.; Schlegel, H. B.; Scuseria, G. E.; Robb, M. A.; Cheeseman, J. R.; Scalmani, G.; Barone, V.; Petersson, G. A.; Nakatsuji, H.; Li, X.; Caricato, M.; Marenich, A.; Bloino, J.; Janesko, B. G.; Gomperts, R.; Mennucci, B.; Hratchian, H. P.; Ortiz, J. V.; Izmaylov, A. F.; Sonnenberg, J. L.; Williams-Young, D.; Ding, F.; Lipparini, F.; Egidi, F.;

doi:10.6342/NTU202403528

Goings, J.; Peng, B.; Petrone, A.; Henderson, T.; Ranasinghe, D.; Zakrzewski, V. G.; J. Gao, N. R.; Zheng, G.; Liang, W.; Hada, M.; Ehara, M.; Toyota, K.; Fukuda, R.; J. Hasegawa, M. I.; Nakajima, T.; Honda, Y.; Kitao, O.; Nakai, H.; Vreven, T.; Throssell, K.; J. A. Montgomery , J.; Peralta, J. E.; Ogliaro, F.; Bearpark, M.; Heyd, J. J.; Brothers, E.; Kudin, K. N.; Staroverov, V. N.; Keith, T.; Kobayashi, R.; Normand, J.; Raghavachari, K.; Rendell, A.; Burant, J. C.; Iyengar, S. S.; Tomasi, J.; Cossi, M.; Millam, J. M.; Klene, M.; Adamo, C.; Cammi, R.; Ochterski, J. W.; Martin, R. L.; Morokuma, K.; Farkas, O.; Foresman, J. B.; Fox, D. J. *Gaussian 09, Revision D.01*, Gaussian, Inc: Wallingford CT, 2009.

263. Sundaram, A.; Venkatasubramanian, V., Parametric Sensitivity and Search-Space Characterization Studies of Genetic Algorithms for Computer-Aided Polymer Design. *Journal of Chemical Information and Computer Sciences* **1998,** 38, (6), 1177-1191.

264. Hsu, H. H.; Huang, C. H.; Lin, S. T., Fully Automated Molecular Design with Atomic Resolution for Desired Thermophysical Properties. *Industrial & Engineering Chemistry Research* **2018,** 57, (29), 9683-9692.

265. Alshehri, A. S.; Gani, R.; You, F., Deep learning and knowledge-based methods for computer-aided molecular design—toward a unified approach: State-of-the-art and future directions. *Computers & Chemical Engineering* **2020,** 141, 107005.

266. Blaschke, T.; Olivecrona, M.; Engkvist, O.; Bajorath, J.; Chen, H., Application of Generative Autoencoder in De Novo Molecular Design. *Molecular Informatics* **2018,** 37, (1-2), 1700123.

267. Lim, J.; Ryu, S.; Kim, J. W.; Kim, W. Y., Molecular generative model based on conditional variational autoencoder for de novo molecular design. *Journal of Cheminformatics* **2018,** 10, (1), 31.

268. Sanchez-Lengeling, B.; Aspuru-Guzik, A., Inverse molecular design using machine learning: Generative models for matter engineering. *Science* **2018,** 361, (6400), 360.

269. de Almeida, A. F.; Moreira, R.; Rodrigues, T., Synthetic organic chemistry driven by artificial intelligence. *Nature Reviews Chemistry* **2019,** 3, (10), 589-604.

270. Gupta, A.; Müller, A. T.; Huisman, B. J. H.; Fuchs, J. A.; Schneider, P.; Schneider, G., Generative Recurrent Networks for De Novo Drug Design. *Molecular Informatics* **2018,** 37, (1-2), 1700111.

271. Blaschke, T.; Arús-Pous, J.; Chen, H.; Margreitter, C.; Tyrchan, C.; Engkvist, O.;

doi:10.6342/NTU202403528

Papadopoulos, K.; Patronov, A., REINVENT 2.0: An AI Tool for De Novo Drug Design. *Journal of Chemical Information and Modeling* **2020,** 60, (12), 5918-5922.

272. Grisoni, F.; Moret, M.; Lingwood, R.; Schneider, G., Bidirectional Molecule Generation with Recurrent Neural Networks. *Journal of Chemical Information and Modeling* **2020,** 60, (3), 1175-1183.

273. Kotsias, P.-C.; Arús-Pous, J.; Chen, H.; Engkvist, O.; Tyrchan, C.; Bjerrum, E. J., Direct steering of de novo molecular generation with descriptor conditional recurrent neural networks. *Nature Machine Intelligence* **2020,** 2, (5), 254-265.

274. D'Souza, S.; Kv, P.; Balaji, S., Training recurrent neural networks as generative neural networks for molecular structures: how does it impact drug discovery? *Expert Opinion on Drug Discovery* **2022,** 17, (10), 1071-1079.

275. Podda, M.; Bacciu, D.; Micheli, A., A Deep Generative Model for Fragment-Based Molecule Generation. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, Silvia, C.; Roberto, C., Eds. PMLR: Proceedings of Machine Learning Research, 2020; Vol. 108, pp 2240--2250.

276. Olivecrona, M.; Blaschke, T.; Engkvist, O.; Chen, H., Molecular de-novo design through deep reinforcement learning. *Journal of Cheminformatics* **2017,** 9, (1), 48.

277. Segler, M. H. S.; Kogej, T.; Tyrchan, C.; Waller, M. P., Generating Focused Molecule Libraries for Drug Discovery with Recurrent Neural Networks. *ACS Central Science* **2018,** 4, (1), 120-131.

278. Merk, D.; Friedrich, L.; Grisoni, F.; Schneider, G., De Novo Design of Bioactive Small Molecules by Artificial Intelligence. *Molecular Informatics* **2018,** 37, (1-2), 1700153.

279. Popova, M.; Isayev, O.; Tropsha, A., Deep reinforcement learning for de novo drug design. *Science Advances* **2018,** 4, (7), eaap7885.

280. Dollar, O.; Joshi, N.; Beck, D. A. C.; Pfaendtner, J., Attention-based generative models for de novo molecular design. *Chemical Science* **2021,** 12, (24), 8362-8372.

281. Gómez-Bombarelli, R.; Wei, J. N.; Duvenaud, D.; Hernández-Lobato, J. M.; Sánchez-Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Adams, R. P.; Aspuru-Guzik, A., Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Central Science* **2018,** 4, (2), 268-276.

282. Kusner, M. J.; Paige, B.; Hernández-Lobato, J. M., Grammar Variational

doi:10.6342/NTU202403528

Autoencoder. In *Proceedings of the 34th International Conference on Machine Learning*, Doina, P.; Yee Whye, T., Eds. PMLR: Proceedings of Machine Learning Research, 2017; Vol. 70, pp 1945--1954.

283. Dai, H.; Tian, Y.; Dai, B.; Skiena, S.; Song, L., Syntax-Directed Variational Autoencoder for Structured Data. In *Sixth International Conference on Learning Representations*, Vancouver Convention Center, Vancouver CANADA, 2018.

284. Kim, H.; Na, J.; Lee, W. B., Generative Chemical Transformer: Neural Machine Learning of Molecular Geometric Structures from Chemical Language via Attention. *Journal of Chemical Information and Modeling* **2021,** 61, (12), 5804-5814.

285. Simonovsky, M.; Komodakis, N., GraphVAE: Towards Generation of Small Graphs Using Variational Autoencoders. In *6th International Conference on Learning Representations (ICLR)*, Vancouver Convention Center, Vancouver, BC, Canada, 2018.

286. Liu, Q.; Allamanis, M.; Brockschmidt, M.; Gaunt, A. L., Constrained Graph Variational Autoencoders for Molecule Design. In *Advances in Neural Information Processing Systems 31 (NeurIPS 2018)*, Palais des Congrès de Montréal, Montréal CANADA, 2018.

287. Jin, W.; Barzilay, R.; Jaakkola, T. In *Junction tree variational autoencoder for molecular graph generation*, International conference on machine learning, 2018; PMLR: 2018; pp 2323-2332.

288. Samanta, B.; De, A.; Jana, G.; Chattaraj, P. K.; Ganguly, N.; Rodriguez, M. G., NeVAE: A Deep Generative Model for Molecular Graphs. *Proceedings of the AAAI Conference on Artificial Intelligence* **2019,** 33, (01), 1110-1117.

289. Lee, M.; Min, K., MGCVAE: Multi-Objective Inverse Design via Molecular Graph Conditional Variational Autoencoder. *Journal of Chemical Information and Modeling* **2022,** 62, (12), 2943-2950.

290. Li, Y.; Tarlow, D.; Brockschmidt, M.; Zemel, R., Gated Graph Sequence Neural Networks. *arXiv e-prints* **2015**, arXiv:1511.05493.

291. Mercado, R.; Rastemo, T.; Lindelöf, E.; Klambauer, G.; Engkvist, O.; Chen, H.; Jannik Bjerrum, E., Graph networks for molecular design. *Machine Learning: Science and Technology* **2021,** 2, (2), 025023.

292. Pham, T.-H.; Xie, L.; Zhang, P., FAME: Fragment-based Conditional Molecular

doi:10.6342/NTU202403528

Generation for Phenotypic Drug Discovery. In *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*, Society for Industrial and Applied Mathematics: 2022; pp 720-728.

293. Zhou, Z.; Kearnes, S.; Li, L.; Zare, R. N.; Riley, P., Optimization of Molecules via Deep Reinforcement Learning. *Scientific Reports* **2019,** 9, (1), 10752.

294. Gao, W.; Mercado, R.; Coley, C. W., Amortized Tree Generation for Bottom-up Synthesis Planning and Synthesizable Molecular Design. *arXiv e-prints* **2021**, arXiv:2110.06389.

295. Bradshaw, J.; Paige, B.; Kusner, M. J.; Segler, M. H. S.; Hernández-Lobato, J. M., Barking up the right tree: an approach to search over molecule synthesis DAGs. *arXiv e-prints* **2020**, arXiv:2012.11522.

296. Wang, M.; Sun, H.; Wang, J.; Pang, J.; Chai, X.; Xu, L.; Li, H.; Cao, D.; Hou, T., Comprehensive assessment of deep generative architectures for de novo drug design. *Briefings in Bioinformatics* **2022,** 23, (1), bbab544.

297. Cai, C.; Wang, S.; Xu, Y.; Zhang, W.; Tang, K.; Ouyang, Q.; Lai, L.; Pei, J., Transfer Learning for Drug Discovery. *Journal of Medicinal Chemistry* **2020,** 63, (16), 8683-8694.

298. Amabilino, S.; Pogány, P.; Pickett, S. D.; Green, D. V. S., Guidelines for Recurrent Neural Network Transfer Learning-Based Molecular Generation of Focused Libraries. *Journal of Chemical Information and Modeling* **2020,** 60, (12), 5699-5713.

299. He, J.; Nittinger, E.; Tyrchan, C.; Czechtizky, W.; Patronov, A.; Bjerrum, E. J.; Engkvist, O., Transformer-based molecular optimization beyond matched molecular pairs. *Journal of Cheminformatics* **2022,** 14, (1), 18.

300. Huang, Y.; Peng, X.; Ma, J.; Zhang, M., 3DLinker: An E(3) Equivariant Variational Autoencoder for Molecular Linker Design. *arXiv e-prints* **2022**, arXiv:2205.07309.

301. Langevin, M.; Minoux, H.; Levesque, M.; Bianciotto, M., Scaffold-Constrained Molecular Generation. *Journal of Chemical Information and Modeling* **2020,** 60, (12), 5637-5646.

302. Polykovskiy, D.; Zhebrak, A.; Sanchez-Lengeling, B.; Golovanov, S.; Tatanov, O.; Belyaev, S.; Kurbanov, R.; Artamonov, A.; Aladinskiy, V.; Veselov, M.; Kadurin, A.; Johansson, S.; Chen, H.; Nikolenko, S.; Aspuru-Guzik, A.; Zhavoronkov, A., Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models. *Frontiers in*

doi:10.6342/NTU202403528

Pharmacology **2020,** 11.

303. Nigam, A.; Pollice, R.; Tom, G.; Jorner, K.; Thiede, L. A.; Kundaje, A.; Aspuru-Guzik, A., Tartarus: A Benchmarking Platform for Realistic And Practical Inverse Molecular Design. *arXiv e-prints* **2022**, arXiv:2209.12487.

304. Gao, W.; Fu, T.; Sun, J.; Coley, C., Sample efficiency matters: a benchmark for practical molecular optimization. *Advances in Neural Information Processing Systems* **2022,** 35, 21342-21357.

305. Wager, T. T.; Hou, X.; Verhoest, P. R.; Villalobos, A., Central Nervous System Multiparameter Optimization Desirability: Application in Drug Discovery. *ACS Chemical Neuroscience* **2016,** 7, (6), 767-775.

306. Bajusz, D.; Rácz, A.; Héberger, K., Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of Cheminformatics* **2015,** 7, (1), 20.

307. Benhenda, M., Can AI reproduce observed chemical diversity? *bioRxiv* **2018**, 292177.

308. Yan, C.; Yang, J.; Ma, H.; Wang, S.; Huang, J., Molecule Sequence Generation with Rebalanced Variational Autoencoder Loss. *Journal of Computational Biology* **2022,** 30, (1), 82-94.

309. Bresson, X.; Laurent, T., A Two-Step Graph Convolutional Decoder for Molecule Generation. In *Thirty-third Conference on Neural Information Processing Systems*, Vancouver Convention Center, Vancouver CANADA 2019.

310. Kajino, H., Molecular Hypergraph Grammar with Its Application to Molecular Optimization. In *Proceedings of the 36th International Conference on Machine Learning*, Kamalika, C.; Ruslan, S., Eds. PMLR: Proceedings of Machine Learning Research, 2019; Vol. 97, pp 3183--3191.

311. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; Polosukhin, I., Attention is all you need. *Advances in Neural Information Processing Systems* **2017,** 30.

312. Wang, W.; Wang, Y.; Zhao, H.; Sciabola, S., A Pre-trained Conditional Transformer for Target-specific De Novo Molecular Generation. *arXiv preprint arXiv:2210.08749* **2022**.

313. Dollar, O.; Joshi, N.; Pfaendtner, J.; Beck, D. A. C., Efficient 3D Molecular Design with an E(3) Invariant Transformer VAE. *The Journal of Physical Chemistry A* **2023,** 127,

doi:10.6342/NTU202403528

(37), 7844-7852.

314. Bagal, V.; Aggarwal, R.; Vinod, P. K.; Priyakumar, U. D., MolGPT: Molecular Generation Using a Transformer-Decoder Model. *Journal of Chemical Information and Modeling* **2022,** 62, (9), 2064-2076.

315. Wang, J.; Hsieh, C.-Y.; Wang, M.; Wang, X.; Wu, Z.; Jiang, D.; Liao, B.; Zhang, X.; Yang, B.; He, Q.; Cao, D.; Chen, X.; Hou, T., Multi-constraint molecular generation based on conditional transformer, knowledge distillation and reinforcement learning. *Nature Machine Intelligence* **2021,** 3, (10), 914-922.

316. Wang, W.; Wang, Y.; Zhao, H.; Sciabola, S., A Pre-trained Conditional Transformer for Target-specific De Novo Molecular Generation *arXiv e-prints* **2022**, arXiv:2210.08749.

317. Gao, W.; Fu, T.; Sun, J.; Coley, C. W., Sample Efficiency Matters: A Benchmark for Practical Molecular Optimization. *arXiv e-prints* **2022**, arXiv:2206.12411.

318. Ciepliński, T.; Danel, T.; Podlewska, S.; Jastrzębski, S., Generative Models Should at Least Be Able to Design Molecules That Dock Well: A New Benchmark. *Journal of Chemical Information and Modeling* **2023,** 63, (11), 3238-3247.

319. Tripp, A.; Simm, G. N. C.; Hernández-Lobato, J. M., A fresh look at de novo molecular design benchmarks. In *NeurIPS 2021 AI for Science Workshop*, 2021.

320. Wu, Z.; Ramsundar, B.; Feinberg, Evan N.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Leswing, K.; Pande, V., MoleculeNet: a benchmark for molecular machine learning. *Chemical Science* **2018,** 9, (2), 513-530.

321. Xu, P.; Feng, T.; Fu, T.; Laghuvarapu, S.; Sun, J., Molecular De Novo Design through Transformer-based Reinforcement Learning. *arXiv e-prints* **2023**, arXiv:2310.05365.

322. Schultheiss, N.; Newman, A., Pharmaceutical Cocrystals and Their Physicochemical Properties. *Crystal Growth & Design* **2009,** 9, (6), 2950-2967.

323. Luo, F.; Liu, X.; Chen, S.; Song, Y.; Yi, X.; Xue, C.; Sun, L.; Li, J., Comprehensive Evaluation of a Deep Eutectic Solvent Based CO2 Capture Process through Experiment and Simulation. *ACS Sustainable Chemistry & Engineering* **2021,** 9, (30), 10250-10265.

324. Hansen, B. B.; Spittle, S.; Chen, B.; Poe, D.; Zhang, Y.; Klein, J. M.; Horton, A.; Adhikari, L.; Zelovich, T.; Doherty, B. W.; Gurkan, B.; Maginn, E. J.; Ragauskas, A.; Dadmun, M.; Zawodzinski, T. A.; Baker, G. A.; Tuckerman, M. E.; Savinell, R. F.; Sangoro, J. R., Deep Eutectic Solvents: A Review of Fundamentals and Applications. *Chemical Reviews* **2021,**

doi:10.6342/NTU202403528

121, (3), 1232-1285.

325. Hung, Y.-C.; Chao, C.-Y.; Dai, C.-A.; Su, W.-F.; Lin, S.-T., Band Gap Engineering via Controlling Donor–Acceptor Compositions in Conjugated Copolymers. *The Journal of Physical Chemistry B* **2013,** 117, (2), 690-696.

326. Makkar, P.; Ghosh, N. N., A review on the use of DFT for the prediction of the properties of nanomaterials. *RSC Advances* **2021,** 11, (45), 27897-27924.

327. Datta, L. P.; Manchineella, S.; Govindaraju, T., Biomolecules-derived biomaterials. *Biomaterials* **2020,** 230, 119633.

328. Venkatasubramanian, V.; Chan, K.; Caruthers, J. M., Computer-aided molecular design using genetic algorithms. *Computers & Chemical Engineering* **1994,** 18, (9), 833-844.

329. Venkatasubramanian, V.; Chan, K.; Caruthers, J. M., Evolutionary design of molecules with desired properties using the genetic algorithm. *J. Chem. Inf. Comput. Sci.* **1995,** 35, (2), 188-195.

330. Venkatasubramanian, V.; Chan, K.; Caruthers, J. M., Genetic Algorithmic Approach for Computer-Aided Molecular Design. In *Computer-Aided Molecular Design*, Reynolds, C. H.; Holloway, M. K.; Cox, H. K., Eds. American Chemical Society: 1995; pp 396-414.

331. Mannodi-Kanakkithodi, A.; Pilania, G.; Huan, T. D.; Lookman, T.; Ramprasad, R., Machine Learning Strategy for Accelerated Design of Polymer Dielectrics. *Sci. Rep.* **2016,** 6, 20952.

332. Evans, R.; O'Neill, M.; Pritzel, A.; Antropova, N.; Senior, A.; Green, T.; Žídek, A.; Bates, R.; Blackwell, S.; Yim, J.; Ronneberger, O.; Bodenstein, S.; Zielinski, M.; Bridgland, A.; Potapenko, A.; Cowie, A.; Tunyasuvunakool, K.; Jain, R.; Clancy, E.; Kohli, P.; Jumper, J.; Hassabis, D., Protein complex prediction with AlphaFold-Multimer. *bioRxiv* **2022**, 2021.10.04.463034.

333. Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Žídek, A.; Potapenko, A.; Bridgland, A.; Meyer, C.; Kohl, S. A. A.; Ballard, A. J.; Cowie, A.; Romera-Paredes, B.; Nikolov, S.; Jain, R.; Adler, J.; Back, T.; Petersen, S.; Reiman, D.; Clancy, E.; Zielinski, M.; Steinegger, M.; Pacholska, M.; Berghammer, T.; Bodenstein, S.; Silver, D.; Vinyals, O.; Senior, A. W.; Kavukcuoglu, K.; Kohli,

doi:10.6342/NTU202403528

P.; Hassabis, D., Highly accurate protein structure prediction with AlphaFold. *Nature* **2021,** 596, (7873), 583-589.

334. Bardow, A.; Steur, K.; Gross, J., Continuous-Molecular Targeting for Integrated Solvent and Process Design. *Industrial & Engineering Chemistry Research* **2010,** 49, (6), 2834-2840.

335. Chen, Y.; Koumaditi, E.; Gani, R.; Kontogeorgis, G. M.; Woodley, J. M., Computer-aided design of ionic liquids for hybrid process schemes. *Computers & Chemical Engineering* **2019,** 130, 106556.

336. Chai, S.; Liu, Q.; Liang, X.; Guo, Y.; Zhang, S.; Xu, C.; Du, J.; Yuan, Z.; Zhang, L.; Gani, R., A grand product design model for crystallization solvent design. *Computers & Chemical Engineering* **2020,** 135, 106764.

337. Chen, Y.; Gani, R.; Kontogeorgis, G. M.; Woodley, J. M., Integrated ionic liquid and process design involving azeotropic separation processes. *Chemical Engineering Science* **2019,** 203, 402-414.

338. Scheffczyk, J.; Schäfer, P.; Fleitmann, L.; Thien, J.; Redepenning, C.; Leonhard, K.; Marquardt, W.; Bardow, A., COSMO-CAMPD: a framework for integrated design of molecules and processes based on COSMO-RS. *Molecular Systems Design & Engineering* **2018,** 3, (4), 645-657.

339. Gertig, C.; Fleitmann, L.; Schilling, J.; Leonhard, K.; Bardow, A., Rx-COSMO-CAMPD: Enhancing Reactions by Integrated Computer-Aided Design of Solvents and Processes based on Quantum Chemistry. *Chemie Ingenieur Technik* **2020,** 92, (10), 1489-1500.

340. III, R. D. J., NIST Computational Chemistry Comparison and Benchmark Database. In May 2022 ed.; 2022.

341. William E. Acree, J.; Chickos, J. S., NIST Chemistry WebBook, NIST Standard Reference Database Number 69. In Linstrom, P. J.; Mallard, W. G., Eds. National Institute of Standards and Technology: Gaithersburg MD, 20899, 2023.

342. Bohacek, R. S.; McMartin, C.; Guida, W. C., The art and practice of structure-based drug design: A molecular modeling perspective. *Medicinal Research Reviews* **1996,** 16, (1), 3-50.

343. Walters, W. P.; Stahl, M. T.; Murcko, M. A., Virtual screening—an overview. *Drug*

doi:10.6342/NTU202403528

Discovery Today **1998,** 3, (4), 160-178.

344. Lemonick, S., Exploring chemical space: can AI take us where no human has gone before? *Chemical & Engineering News* **2020,** 98, (13), 30-35.

345. Dyk, B. v.; Nieuwoudt, I., Design of Solvents for Extractive Distillation. *Ind. Eng. Chem. Res.* **2000,** 39, (5), 1423-1429.

346. Wu, L.-L.; Chang, W.-X.; Guan, G.-F., Extractants Design Based on an Improved Genetic Algorithm. *Ind. Eng. Chem. Res.* **2007,** 46, (4), 1254-1258.

347. Heintz, J.; Belaud, J.-P.; Pandya, N.; Teles Dos Santos, M.; Gerbaud, V., Computer aided product design tool for sustainable product development. *Comput. Chem. Eng.* **2014,** 71, (Supplement C), 362-376.

348. Zhou, T.; Wang, J.; McBride, K.; Sundmacher, K., Optimal design of solvents for extractive reaction processes. *AIChE J.* **2016,** 62, (9), 3238-3249.

349. Zhou, T.; Zhou, Y.; Sundmacher, K., A hybrid stochastic–deterministic optimization approach for integrated solvent and process design. *Chemical Engineering Science* **2017,** 159, 207-216.

350. Karunanithi, A. T.; Mehrkesh, A., Computer-aided design of tailor-made ionic liquids. *Aiche Journal* **2013,** 59, (12), 4627-4640.

351. Glen, R. C.; Payne, A. W. R., A genetic algorithm for the automated generation of molecules within constraints. *J. Comput.-Aided Mol. Des.* **1995,** 9, (2), 181-202.

352. Douguet, D.; Thoreau, E.; Grassy, G., A genetic algorithm for the automated generation of small organic molecules: drug design using an evolutionary algorithm. *J Comput Aided Mol Des* **2000,** 14, (5), 449-66.

353. Kamphausen, S.; Höltge, N.; Wirsching, F.; Morys-Wortmann, C.; Riester, D.; Goetz, R.; Thürk, M.; Schwienhorst, A., Genetic algorithm for the design of molecules with desired properties. *J. Comput.-Aided Mol. Des.* **2002,** 16, (8), 551-567.

354. Douguet, D.; Munier-Lehmann, H.; Labesse, G.; Pochet, S., LEA3D: A Computer-Aided Ligand Design for Structure-Based Drug Design. *J. Med. Chem.* **2005,** 48, (7), 2457-2468.

355. Dey, F.; Caflisch, A., Fragment-based de novo ligand design by multiobjective evolutionary optimization. *J Chem Inf Model* **2008,** 48, (3), 679-90.

356. Scheffczyk, J.; Fleitmann, L.; Schwarz, A.; Lampe, M.; Bardow, A.; Leonhard, K.,

doi:10.6342/NTU202403528

COSMO-CAMD: A framework for optimization-based computer-aided molecular design using COSMO-RS. *Chem. Eng. Sci.* **2017,** 159, 84-92.

357. Struebing, H.; Obermeier, S.; Siougkrou, E.; Adjiman, C. S.; Galindo, A., A QM-CAMD approach to solvent design for optimal reaction rates. *Chem. Eng. Sci.* **2017,** 159, 69-83.

358. Kirkpatrick, S.; Jr., C. D. G.; Vecchi, M. P., Optimization by Simulated Annealing. *Science* **1983,** 220, (4598), 671-680.

359. Sorkin, G. B., Efficient Simulated Annealing on Fractal Energy Landscapes. *Algorithmica* **1991,** 6, 367-418.

360. Cardoso, M. E.; Salcedo, R. L.; Azevedo, S. F. d.; Barbosa, D., A simulated annealing approach to the solution of minlp problems. *Comput. Chem. Eng.* **1997,** 21, 1349-1364.

361. Dekkers, A.; Aarts, E., Global optimization and simulated annealing. *Math. Program.* **1991,** 50, (1-3), 367-393.

362. Kim, K.-J.; Diwekar, U. M., Efficient Combinatorial Optimization under Uncertainty. 1. Algorithmic Development. *Ind. Eng. Chem. Res.* **2002,** 41, (5), 1276-1284.

363. Faulon, J.-L., Stochastic Generator of Chemical Structure. 2. Using Simulated Annealing To Search the Space of Constitutional Isomers. *J. Chem. Inf. Comput. Sci.* **1996,** 36, (4), 731-740.

364. Marcoulaki, E. C.; Kokossis, A. C., Molecular Design Synthesis Using Stochastic Optimisation as a Tool for Scoping and Screening. *Computers and Chemical Engineering* **1998,** 22, S11-S18.

365. Ourique, J. E.; Silva Telles, A., Computer-aided molecular design with simulated annealing and molecular graphs. *Comput. Chem. Eng.* **1998,** 22, (Supplement 1), S615-S618.

366. Kim, K.-J.; Diwekar, U. M., Efficient Combinatorial Optimization under Uncertainty. 2. Application to Stochastic Solvent Selection. *Ind. Eng. Chem. Res.* **2002,** 41, (5), 1285-1296.

367. Kim, K.-J.; Diwekar, U. M., Hammersley stochastic annealing: efficiency improvement for combinatorial optimization under uncertainty. *IIE Trans.* **2002,** 34, (9), 761-777.

368. Papadopoulos, A. I.; Linke, P., Multiobjective molecular design for integrated

doi:10.6342/NTU202403528

process-solvent systems synthesis. *AIChE J.* **2006,** 52, (3), 1057-1070.

369. Liu, B.; Wen, Y.; Zhang, X., Development of CAMD based on the hybrid gene algorithm and simulated annealing algorithm and the application on solvent selection. *Can. J. Chem. Eng.* **2017,** 95, (4), 767-774.

370. Zhang, J.; Qin, L.; Peng, D.; Zhou, T.; Cheng, H.; Chen, L.; Qi, Z., COSMO-descriptor based computer-aided ionic liquid design for separation processes: Part II: Task-specific design for extraction processes. *Chem. Eng. Sci.* **2017,** 162, 364-374.

371. Zhang, J.; Peng, D.; Song, Z.; Zhou, T.; Cheng, H.; Chen, L.; Qi, Z., COSMO-descriptor based computer-aided ionic liquid design for separation processes. Part I: Modified group contribution methodology for predicting surface charge density profile of ionic liquids. *Chem. Eng. Sci.* **2017,** 162, 355-363.

372. Diwekar, U. M.; Gebreslassie, B. H., Efficient ant colony optimization (EACO) algorithm for deterministic optimization. *Int. J. Swarm Intel. Evol. Comput.* **2016,** 5, (131).

373. Gebreslassie, B. H.; Diwekar, U. M., Efficient ant colony optimization for computer aided molecular design: Case study solvent selection problem. *Comput. Chem. Eng.* **2015,** 78, 1-9.

374. Lin, B.; Chavali, S.; Camarda, K.; Miller, D. C., Computer-aided molecular design using Tabu search. *Comput. Chem. Eng.* **2005,** 29, (2), 337-347.

375. McLeese, S. E.; Eslick, J. C.; Hoffmann, N. J.; Scurto, A. M.; Camarda, K. V., Design of ionic liquids via computational molecular design. *Comput. Chem. Eng.* **2010,** 34, (9), 1476-1480.

376. Vaidyanathan, R.; El-Halwagi, M., Computer-Aided Synthesis of Polymers and Blends with Target Properties. *Ind. Eng. Chem. Res.* **1996,** 35, (2), 627-634.

377. Roughton, B. C.; Christian, B.; White, J.; Camarda, K. V.; Gani, R., Simultaneous design of ionic liquid entrainers and energy efficient azeotropic separation processes. *Comput. Chem. Eng.* **2012,** 42, 248-262.

378. Camarda, K. V.; Maranas, C. D., Optimization in Polymer Design Using Connectivity Indices. *Ind. Eng. Chem. Res.* **1999,** 38, (5), 1884-1892.

379. Gopinath, S.; Jackson, G.; Galindo, A.; Adjiman, C. S., Outer approximation algorithm with physical domain reduction for computer-aided molecular and separation process design. *AIChE J.* **2016,** 62, (9), 3484-3504.

doi:10.6342/NTU202403528

380. Wang, Y.; Achenie, L. E. K., Computer aided solvent design for extractive fermentation. *Fluid Phase Equilib.* **2002,** 201, (1), 1-18.

381. Maranas, C. D., Novel Mathematical Programming Model for Computer Aided Molecular Design. *Ind. Eng. Chem. Res.* **1996,** 35, (10), 3403-3414.

382. Vaidyanathan, R.; El-Halwagi, M., Global Optimization of Nonconvex MINLP's by Interval Analysis. In *Global Optimization in Engineering Design*, Grossmann, I. E., Ed. Springer US: Boston, MA, 1996; Vol. 9, pp 175-193.

383. Ryoo, H. S.; Sahinidis, N. V., A branch-and-reduce approach to global optimization. *J. Global Optim.* **1996,** 8, (2), 107-138.

384. Sahinidis, N. V.; Tawarmalani, M.; Yu, M., Design of alternative refrigerants via global optimization. *AIChE J.* **2003,** 49, (7), 1761-1775.

385. Samudra, A. P.; Sahinidis, N. V., Optimization-based framework for computer-aided molecular design. *AIChE J.* **2013,** 59, (10), 3686-3701.

386. Brignole, E. A.; Bottini, S.; Gani, R., A Strategy for The Design and Selection of Solvent for Seperation Processes. *Fluid Phase Equilib.* **1986,** 29, (Supplement C), 125-132.

387. Gani, R.; Nielsen, B.; Fredenslund, A., A group contribution approach to computer-aided molecular design. *Aiche Journal* **1991,** 37, (9), 1318-1332.

388. Pretel, E. J.; López, P. A.; Bottini, S. B.; Brignole, E. A., Computer-aided molecular design of solvents for separation processes. *AIChE J.* **1994,** 40, (8), 1349-1360.

389. Harper, P. M.; Gani, R.; Kolar, P.; Ishikawa, T., Computer-aided molecular design with combined molecular modeling and group contribution. *Fluid Phase Equilib.* **1999,** 158-160, 337-347.

390. Hostrup, M.; Harper, P. M.; Gani, R., Design of environmentally benign processes: integration of solvent design and separation process synthesis. *Comput. Chem. Eng.* **1999,** 23, (10), 1395-1414.

391. Karunanithi, A. T.; Achenie, L. E. K.; Gani, R., A computer-aided molecular design framework for crystallization solvent design. *Chem. Eng. Sci.* **2006,** 61, (4), 1247-1260.

392. Zhang, L.; Cignitti, S.; Gani, R., Generic mathematical programming formulation and solution for computer-aided molecular design. *Comput. Chem. Eng.* **2015,** 78, 79-84.

393. Nocedal, J.; Wright, S. J., *Numerical optimization*. Springer: 1999.

394. Bazaraa, M. S.; Sherali, H. D.; Shetty, C. M., *Nonlinear programming: theory and*

doi:10.6342/NTU202403528

algorithms. John wiley & sons: 2006.

395. Andrei, N., *Modern Numerical Nonlinear Optimization*. Springer: 2022; Vol. 195.

396. Cybenko, G., Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems* **1989,** 2, (4), 303-314.

397. Goodfellow, I.; Bengio, Y.; Courville, A., *Deep learning*. MIT press: 2016.

398. Scarff, B. Understanding Backpropagation. https://towardsdatascience.com/understanding-backpropagation-abcc509ca9d0 (2023/6/20),

399. Rumelhart, D. E.; Hinton, G. E.; Williams, R. J., Learning representations by back-propagating errors. *Nature* **1986,** 323, (6088), 533-536.

400. Gers, F. A.; Schmidhuber, E., LSTM recurrent networks learn simple context-free and context-sensitive languages. *IEEE Transactions on Neural Networks* **2001,** 12, (6), 1333-1340.

401. Goldberg, Y., A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research* **2016,** 57, 345-420.

402. Eck, D.; Schmidhuber, J. In *Finding temporal structure in music: blues improvisation with LSTM recurrent networks*, Proceedings of the 12th IEEE Workshop on Neural Networks for Signal Processing, 6-6 Sept. 2002, 2002; 2002; pp 747-756.

403. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y., Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* **2014**.

404. Guo, C.; Berkhahn, F., Entity embeddings of categorical variables. *arXiv preprint arXiv:1604.06737* **2016**.

405. Culurciello, E. The fall of RNN / LSTM. https://towardsdatascience.com/the-fall-of-rnn-lstm-2d1594c74ce0 (2023/06/20),

406. Razvan, P.; Tomas, M.; Yoshua, B., On the difficulty of training recurrent neural networks. In PMLR: 2013; Vol. 28, pp 1310-1318.

407. Kafunah, J. Vanishing And Exploding Gradient Problems. https://www.jefkine.com/general/2018/05/21/2018-05-21-vanishing-and-exploding-gradient-problems/ (2023/6/20),

408. Arbel, N. How LSTM networks solve the problem of vanishing gradients: A simple, straightforward mathematical explanation.

doi:10.6342/NTU202403528

https://medium.datadriveninvestor.com/how-do-lstm-networks-solve-the-problem-of-vanishing-gradients-a6784971a577 (2023/6/20),

409. Ribeiro, A. H.; Tiels, K.; Aguirre, L. A.; Schön, T., Beyond exploding and vanishing gradients: analysing RNN training using attractors and smoothness. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, Silvia, C.; Roberto, C., Eds. PMLR: Proceedings of Machine Learning Research, 2020; Vol. 108, pp 2370--2380.

410. Williams, R. J.; Zipser, D., A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Computation* **1989,** 1, (2), 270-280.

411. Spinner, T.; Körner, J.; Görtler, J.; Deussen, O. In *Towards an interpretable latent space: an intuitive comparison of autoencoders with variational autoencoders*, IEEE VIS 2018, 2018; 2018.

412. Sadegh, M.; Bing, O. D.; Christian, P.-E.; Linus, G., *Penalized Variational Autoencoder for Molecular Design*. 2019.

doi:10.6342/NTU202403528