Machine learning for de novo design of functional molecules and their synthetic routes

Zhang Qi

Department of Statistical Science
School of Multidisciplinary Sciences
The Graduate University for Advanced Studies, SOKENDAI

This dissertation is submitted for the degree of
Doctor of Philosophy

September 2023

Abstract

In recent years, de novo molecular design using machine learning has made significant technical progress, but practical applications still face challenges. The primary barrier to practical applications of computationally designed molecules is the cost and technical difficulty of synthesizing them. As a solution to this problem, various methods for synthetic route design using deep neural networks have been studied. However, designing molecules and their synthetic routes simultaneously has received little attention. In this study, we formulate the problem of designing molecules and their synthetic routes within the framework of Bayesian inference. The design variables in our framework consist of a set of reactants in a reaction network and its network topology. The design space is vast, consisting of all combinations of purchasable reactants, which can number in the millions or more. In addition, the designed reaction networks can adopt any topology beyond simple multistep linear reaction routes. Thus, this is a hard combinatorial problem that requires a powerful algorithm to solve. To address this problem, we present a powerful sequential Monte Carlo algorithm that recursively designs a synthetic reaction network by sequentially building up single-step reactions. Our algorithm is highly efficient, and in a case study of designing drug-like molecules based on commercially available compounds, it outperformed heuristic combinatorial search methods in terms of computational efficiency, coverage, and novelty with respect to existing compounds. Our framework has practical implications in the design of new molecules for various applications, such as drug design and materials science. Moreover, it offers a significant advantage in terms of computational efficiency, which is crucial for large-scale molecular design. Our study also provides the Python library "Seq-Stack-Reaction" with an illustrative example of designing highly viscous lubricant molecules. This library can be used to design molecules with desired properties and synthetic routes, making it a valuable tool for researchers in various fields.

Acknowledgements

I would like to express my heartfelt gratitude to everyone who has helped and supported me during the course of my research. This research project would not have been possible without the guidance and support of many people, and I feel honored to have had the opportunity to work with such a talented and dedicated team. First and foremost, I would like to thank my supervisor, Professor Ryo, for his invaluable guidance and support throughout this project. His knowledge, experience, and expertise in the field of research have been an invaluable resource for me, and his unwavering support and encouragement have been instrumental in my success. I am deeply grateful for the time and effort he invested in mentoring me, and for his unwavering commitment to my development as a researcher.

I would like to express my sincerest gratitude to the examiner Daichi Mochihashi, Kei Terayama, Shin'ya Nakano, and Stephen Wu for their dedicated attendance at my examination and for providing invaluable insights and comments. I extend my heartfelt appreciation to them for their time, effort, and dedication. Their presence and thoughtful observations have been instrumental in shaping the quality and depth of my work.

I would also like to extend my gratitude to the lab members, who have been an integral part of this project. Their contributions and feedback have been invaluable in shaping the direction and focus of my research. I would like to thank Chang Liu, Stephen Wu, and Yoshihiro Hayashi, for their assistance in conducting experiments and analyzing data. Their technical expertise and support have been indispensable in successfully completing this project.

Finally, I would like to express my heartfelt appreciation to my family and friends, who have been a constant source of love and support throughout this project. Their encouragement and understanding have been instrumental in keeping me motivated and focused, and their unwavering belief in me has been a constant source of inspiration.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent years, machine learning-based molecular design has made great technological advances and has achieved remarkable outcomes in drug discovery and materials science (Ikebata et al., 2017; Wu et al., 2019; Gómez-Bombarelli et al., 2018; Segler et al., 2018a; Sanchez-Lengeling & Aspuru-Guzik, 2018; Ramprasad et al., 2017; Perron et al., 2022; Sumita et al., 2022). The design objective here is to identify the chemical structure of a new molecule with a given set of desired properties. To do so, a machine learning model that forwardly predicts the physicochemical or other properties of any given chemical structure is first constructed and then its inverse mapping is found to determine the design of the chemical structure that exhibits the desired properties in the backward direction. The former is often referred to as quantitative structure–property relationship (QSPR) analysis (Roy et al., 2015), and the latter is called inverse structure–property relationship (inverse-QSPR) analysis (Miyao et al., 2016). Usually, the inverse problem is solved using heuristic search techniques such as a genetic algorithm. A molecular generator is then used to sequentially modify a candidate molecule such that the resulting predicted properties fall into the region of the desired properties. The generative model plays an important role in this process. Traditionally, structural modifications have been performed by conducting stochastic recombination with a predefined set of molecular fragments or random atomic substitution. By utilizing fragments of existing molecules as building blocks, we can restrict the degrees of freedom in the resulting chemical structures and narrow down the search space to enhance the synthesizability of virtually created molecules. However, excessive narrowing of the search space may reduce the novelty of the structures created. In order to overcome this limitation, increasing attention has been paid, since around 2017–2018 in particular, to the development of molecule generative models that rely on advanced machine learning techniques (Ikebata et al., 2017; Yang et al., 2017; Assouel et al., 2018; Dai et al., 2018; Gómez-Bombarelli et al., 2018; Jin et al., 2018; Kadurin et al., 2017; Kajino, 2019; Li et al., 2018; Seff et al., 2019;

Segler et al., 2018a; Simm & Hernández-Lobato, 2019; Simm et al., 2020; You et al., 2018). With a training set of compounds synthesized thus far, a generative model for molecular graphs is constructed to mimic the rule of frequently appearing chemical fragments and bonding patterns in the training molecules. Using such a model, we can freely scan the vast chemical space to identify innovative hypothetical molecules.

As described above, machine learning-based molecular design has made great technical progress over the past few years. However, there are still hurdles to be overcome in their practical applications. One reason for this is owing to the difficulty in determining the synthetic routes to such designed molecules. Concurrently, with the growing attention to molecular design, significant progress has been made in computational methods for synthetic route design (Schwaller et al., 2019; Jin et al., 2017; Liu et al., 2017; Lin et al., 2020; Zheng et al., 2019; Nguyen & Tsuda, 2022; Segler et al., 2018b). Similar to the molecular design task, the general workflow for synthetic route design consists of forward and inverse problems. The goal of the forward problem is to derive a model that predicts the chemical structure of a synthetic product for a given set of reactant molecules. In contrast, in the inverse problem, inverse mapping of the forward model is explored to identify a set of reactants that produces a given desired product. Recent developments in deep learning technologies have significantly improved the accuracy of predicting reaction outcomes in organic synthesis. For example, if the structures of the reactants and products are treated as graphs, the reaction prediction task can be formulated as a graph transformation problem (Jin et al., 2017). Under the string representation of reactants and products according to the simplified molecular input line entry system (SMILES) chemical language (Weininger, 1988), deep neural networks for sequence-to-sequence translation can be utilized to predict the SMILES string of a product from input reactants. For example, it was reported that the reaction prediction using Transformer, a well-known encoder-decoder architecture for machine translation, could successfully predict the chemical structures of synthetic products with over 90% accuracy (Schwaller et al., 2019). Furthermore, several methods have been developed to solve the inverse problem of such forward reaction prediction models. The objective is to identify a set of promising reactants from a list of commercially available compounds that can synthesize a desired product (Guo et al., 2020; Bradshaw et al., 2020; Gottipati et al., 2020).

More recently, several attempts have been made to simultaneously design desired functional molecules and their synthetic routes under a unified methodological framework. For a given reaction prediction model, a virtual library of candidate molecules can be created by which a set of reactants is given to the model to produce its synthetic product. In addition, the properties or a certain score of a candidate molecule can be calculated using machine learning models or an arbitrary reward function. By obtaining the inverse mapping of such a cascade model, which defines the composite

mapping from any given reactant set to a product and from the product to its properties, the products exhibiting desired properties and their reactant sets can be predicted simultaneously. The Molecule Chef algorithm proposed by Bradshaw et al. (2019) used a cascaded forward model that connects a deep generative model of reactant molecules (Molecular Transformer) for the reaction prediction (Schwaller et al., 2019), and a property prediction model for the created virtual products. Gottipati et al. (2020) formulated the inverse problem of such a cascade model as a combinatorial optimization problem over a set of commercially available compounds, and proposed a reinforcement learning algorithm to identify the promising combination of reactant molecules. Gao et al. (2021) formulated the problem of synthesis planning as generating a synthetic tree, and proposed a Markov decision process method to construct the synthetic tree in a bottom-up manner. Horwood & Noutahi (2020) considered a reinforcement learning framework for molecular design to use known chemistry as a starting point and optimize it by sequentially performing reactions. However, there are still many unsolved technical problems in these existing methods, and their predictive performance remains yet to be improved. One of the technical difficulties comes from the complexity of the search space. The search space grows exponentially due to the degree of freedom in the network topology of designed synthetic pathways and the possible combinations of reactants. Therefore, it should be theoretically guaranteed that a given computational method can reach the entire search space. The computational burden of exploring such a large space is another important issue; more attention should be paid to developing methods that accelerate the search process. This study aims to provide solutions to these problems.

In this paper, we formulate the task of simultaneous design of molecules and synthetic routes as a general statistical problem in which a forward model is defined as a cascade of reaction prediction models and property prediction models, as in the previous work of Gottipati et al. (2020). In the inverse problem, we seek a set of reactants and a network structure of synthetic reactions such that the resulting products reach a predefined region of desired properties. The overall search space for the reactant sets is spanned by all possible combinations of given commercially available compounds. If the number of commercial compounds is of the order $O(10^6)$ and the number of reactants involved in a synthetic route is 10, the size of the search space will reach $O(10^{60})$. The structure of the reaction network is a design variable as well, involving the number of reaction steps and the network topology.

For example, a pattern of branching routes, the width and depth of networks, and the number of leaf nodes are included in the design variable. Therefore, it is necessary to solve the intractable combinatorial problem defined over an extremely vast search space.

Here, we present an efficient algorithmic technique and implementation to solve this hard problem. Specifically, we present a sequential Monte Carlo method based on

a recurrent network search algorithm to simultaneously identify a reaction network, its constituent reactant sets, and the final products that satisfy the arbitrary target properties. An important practical requirement considered here is the maintenance of diversity in the designed molecules and synthetic routes. There are always errors in the properties and synthetic routes predicted by statistical models. Therefore, the optimal solution in a model is not optimal in practice. Our method aims to exhaustively identify a wide variety of promising candidates and to present various scenarios to domain experts. The final decision is left to the expert. A Python library called Seq-Stack-Reaction has been made available on GitHub (Zhang, 2022), which allows us to plug-in any forward model into the Bayesian design workflow (Figure 1.1).

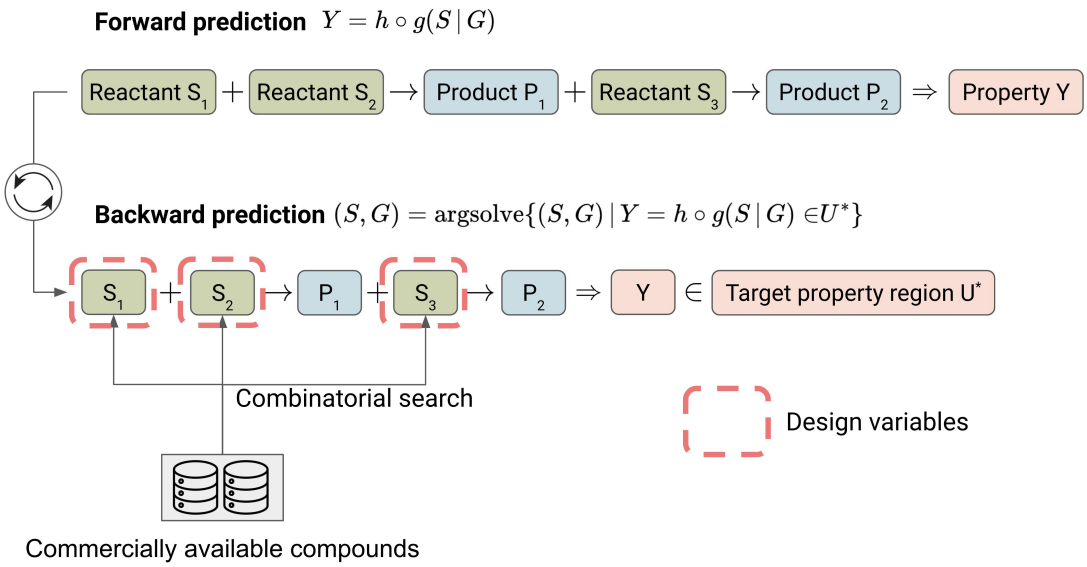


Figure 1.1: Workflow of the concurrent design of molecules and their synthetic reaction networks. The forward-prediction model defines the composite mapping $h \circ g$ from a given reactant set S to its synthetic product P conditioning on a reaction network G via a reaction function g, and from product P to its properties Y via a prediction model h. The desirable products and their synthetic reactions are concurrently designed by inversely exploring G and S such that the resulting properties meet the design objective $Y \in U^*$.

This thesis proceeds as follows.

Chapter 2 Chapter 2 provides an in-depth exploration of the major topics related to machine learning in molecular design. Specifically, the chapter focuses on three key building blocks of molecular design: representation of molecules, deep learning

models, and design objectives. One of the fundamental aspects of molecular design is the representation of molecules. This is because the representation plays a key role in determining the properties and behavior of a molecule. In this chapter, we explore the various methods for molecular representation, including graph-based and sequence-based approaches. Graph-based approaches involve representing a molecule as a graph, where each node represents an atom and each edge represents a bond. Sequence-based approaches, on the other hand, involve representing a molecule as a sequence of characters, such as SMILES or InChI notation. We discuss the strengths and weaknesses of each approach and how they can be used to train machine-learning models for molecular design. The second building block of molecular design that we explore in Chapter 2 is deep learning models. Deep learning has emerged as a powerful tool for machine learning in recent years, and it has been applied to a wide range of tasks in molecular design, such as predicting properties, generating molecules, and optimizing molecular structures. We provide an overview of the various types of deep learning models used in molecular design, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and generative adversarial networks (GANs). We also discuss the advantages and limitations of each type of model and their potential applications in molecular design. Finally, in Chapter 2, we examine the design objectives in molecular design. The ultimate goal of molecular design is to create molecules with desired properties, such as efficacy, safety, and stability. To achieve this, machine learning models must be trained to optimize specific design objectives, such as maximizing or minimizing certain properties. We explore the various types of design objectives used in molecular design, including property prediction, property optimization, and structure generation. We also discuss the challenges and opportunities in optimizing these objectives and how they can be used to develop effective machine-learning models for molecular design.

Chapter 3 Chapter 3 of this study provides a detailed description of the methods that will be used in the proposed methodology for simultaneous molecular design and synthetic pathway design. The chapter is divided into four sections, each discussing a key aspect of the proposed method. The first section introduces the forward model of reaction prediction, which is a critical component of the proposed methodology. The forward model is designed to predict the outcome of a chemical reaction based on the properties of the reactants and the reaction conditions. This section provides an overview of the methodology used to train the model, including the choice of training data and the selection of features. It also discusses the performance of the forward model in predicting the outcome of chemical reactions and its limitations. The second section introduces the forward model of property prediction, which is another key component of the proposed methodology. The forward model is designed to predict the properties of a molecule based on its chemical structure. This section provides an

overview of the methodology used to train the model, including the choice of training data and the selection of features. It also discusses the performance of the forward model in predicting the properties of molecules and its limitations. The third section introduces the Bayesian inverse problem, which will be used for inverse design. The Bayesian inverse problem is a mathematical framework for solving inverse problems, which are problems where the input is known but the output is unknown. In the context of molecular design, the Bayesian inverse problem can be used to identify the optimal set of reactants and reaction conditions that will produce a molecule with a desired set of properties. This section provides an overview of the Bayesian inverse problem and its application to molecular design. The fourth section introduces sequential Monte Carlo, which is a specific method used to solve the inverse problem. Sequential Monte Carlo is a computational technique for approximating the posterior distribution of a system given a set of observations. In the context of molecular design, sequential Monte Carlo can be used to identify the optimal set of reactants and reaction conditions that will produce a molecule with a desired set of properties. This section provides an overview of the sequential Monte Carlo algorithm and its application to molecular design.

Chapter 4 This chapter outlines the proposed method that aims to solve the inverse problem of synthetic reaction network design. The chapter begins by introducing the algorithm used to accomplish this task. Subsequently, an acceleration technique is presented that enhances the efficiency of the proposed method. Finally, the proposed method is summarized as an algorithm, and the software to implement it is described. To begin with, the proposed method employs an algorithm that allows for the simultaneous identification of a reaction network, its constituent reactant sets, and the final products that satisfy arbitrary target properties. The algorithm leverages a synthetic reaction prediction model and property prediction models as building blocks to construct a forward model. The inverse mapping of this forward model is then obtained based on a Bayesian inference framework. As the design space, consisting of arbitrary reaction networks and reactant sets, is exceedingly large, the proposed method adopts a sequential Monte Carlo algorithm incorporating a recurrent algorithm for network search. This approach enables the proposed method to achieve an efficient and effective search for high-quality virtual molecules.
In addition to the algorithm for solving the inverse problem of synthetic reaction network design, an acceleration technique is presented in the proposed method. This technique accelerates the network search process and reduces the computational costs associated with the proposed method. Specifically, the proposed technique involves the use of a surrogate model, which is a simplified version of the forward model that provides an estimate of the forward model's response. This surrogate model is updated iteratively by incorporating the results of each search step, leading to faster

convergence towards the optimal solution.
Finally, the proposed method is summarized as an algorithm, along with the software required to implement it. The algorithm includes several steps, such as initializing the search process, sampling initial reactant sets, and iteratively updating the surrogate model and the search path. The software implementation of the proposed method is designed to be user-friendly and flexible, allowing users to plug-in arbitrary reaction prediction models, property prediction models, and commercial compounds.

Chapter 5 In this chapter, we evaluated the performance of the proposed method in designing drug-like molecules by assessing its predictive and computational capabilities. To accomplish this, we constructed various versions of Bayesian molecular design algorithms by integrating the four constituent mechanisms discussed in the previous section. The aim of this evaluation was to quantitatively investigate the contributions of each mechanism to the overall performance of the algorithm. To start, we selected drug-like molecules as our application example, which are an important class of molecules due to their significant role in the pharmaceutical industry. We then constructed multiple versions of the Bayesian molecular design algorithm that incorporated the following four constituent mechanisms: synthetic reaction prediction, property prediction, Bayesian inverse design, and sequential Monte Carlo. Next, we evaluated the performance of each version of the algorithm in terms of its predictive and computational capabilities. Specifically, we examined the algorithm's ability to design molecules with desirable properties while also considering the complexity and computational efficiency of the algorithm. Through our evaluation, we were able to quantitatively analyze the individual contributions of each constituent mechanism to the overall performance of the algorithm. By comparing the performance of each version of the algorithm, we identified the strengths and weaknesses of each mechanism and determined their relative importance in achieving the desired design objectives.

Chapter 6 In the final chapter, the proposed work is summarized, and the future direction of the research is discussed.

Chapters 3 to 5 are based on our work published work (Zhang et al., 2023) on Science and Technology of Advanced Materials: Methods.

Chapter 2

Machine learning in molecular design

The subject of machine learning has emerged as a vital and rapidly evolving area in the realm of computer-aided molecular design. In contrast to traditional physical models that rely on explicit physical equations, such as quantum chemistry or molecular dynamics simulations, machine learning approaches utilize pattern recognition algorithms to discern mathematical relationships between empirical observations of small molecules, enabling them to extrapolate and predict the chemical, biological, and physical properties of novel compounds. Machine learning techniques are highly efficient and can easily be scaled to accommodate large datasets without the need for extensive computational resources. As such, they are increasingly being utilized to help researchers understand and exploit the intricate relationships between chemical structures and their biological activities or structure-activity relationships (SAR). For example, in the context of a drug screening campaign, researchers may wish to optimize a hit compound's chemical structure to improve its binding affinity, biological response, or physiochemical properties. In the past, this type of problem could only be addressed through numerous costly, time-consuming, and labor-intensive cycles of medicinal chemistry synthesis and analysis. In contrast, modern machine learning techniques can be used to model quantitative structure-property relationships (QSPR) (Puri et al., 2016) and develop artificial intelligence programs that accurately predict in silico how chemical modifications might influence biological behavior. In particular, QSAR modeling has been highly successful in effectively modeling many physiochemical properties of drugs, including toxicity, metabolism, drug-drug interactions, and carcinogenesis. Early QSAR models, such as the Hansch and Free-Wilson analyses, utilized simple multivariate regression models to correlate potency with substructure motifs and chemical properties such as solubility (logP), hydrophobicity, substituent pattern, and electronic factors. However, these approaches were ultimately limited

by the unavailability of experimental data and the linearity assumption made in modeling. Thus, advanced chemoinformatics and machine learning techniques capable of modeling nonlinear datasets, as well as large and complex datasets, are necessary to advance the field of drug discovery. These new methods are expected to offer increased accuracy and efficiency, thereby contributing to the development of novel therapeutics with improved efficacy and fewer adverse side effects.
In this chapter, we delve into the three main topics involved in machine learning-based molecular design, exploring the fundamental concepts and techniques that enable this exciting field. The first topic we review is the representation of molecules. A machine learning model needs to have access to as much information as possible about a chemical structure in order to generate accurate predictions. We explore the different methods of molecular representation, including SMILES notation and molecular fingerprints, and evaluate their strengths and weaknesses. The second topic we delve into is the machine learning model used to learn features from existing data. The selection of the model used is crucial, as it determines the accuracy and efficiency of the molecular design process. We analyze various models, such as neural networks and decision trees, and determine which one is best suited to our needs. The last topic we explore is the objective of the designing task. In this stage, we instruct the machine learning model on what we want it to achieve and measure its performance against specific criteria. We discuss the importance of defining the objective precisely to ensure that the model generates molecules that meet our specific requirements.
Throughout the chapter, we provide examples of how these concepts are applied in real-world scenarios. We showcase the potential of machine learning-based molecular design, including its ability to accelerate drug discovery and design new materials with unique properties. By the end of this chapter, readers will have a solid understanding of the fundamental concepts involved in machine learning-based molecular design, and be able to apply them to their own research in this exciting and rapidly evolving field.

2.1 Representation of molecule

Chemical descriptors are numerical features extracted from chemical structures and are widely used for molecular data mining, compound diversity analysis, and compound activity prediction. In this thesis, we provide a comprehensive overview of chemical descriptors, including their different types and applications.
One-dimensional (1D) descriptors are scalar values that describe aggregate information such as atom counts, bond counts, molecular weight, sums of atomic properties, or fragment counts. While simple to compute, 1D descriptors are limited by their degeneracy problems. In other words, distinct compounds may be mapped to identical descriptor values for a given descriptor. To address this, 1D descriptors are often

used in combination with higher-dimensional descriptors or expressed as a vector of multiple 1D descriptors.
Two-dimensional (2D) descriptors are the most frequently reported descriptor type in the literature. They include topological indices, molecular profiles, and 2D autocorrelation descriptors. One important feature of 2D descriptors is graph invariance, which allows for structure differentiation. This means that descriptor values are unaffected by the renumbering of graph nodes. To facilitate the analysis of large spaces of 2D descriptors, Hong et al. (2008) developed the Mol^2 system, which rapidly generates up to 200 types of 2D descriptors for large compound datasets. Commercial software packages, such as the DRAGON system (Sawada et al., 2014), can also generate up to 5000 types of descriptors as part of several QSAR studies.
Three-dimensional (3D) descriptors extract chemical features from 3D coordinate representations and are considered the most sensitive to structural variations. Common 3D descriptors include autocorrelation descriptors, substituent constants, surface: volume descriptors, and quantum-chemical descriptors. Which are distinct chemical scaffolds with similar binding activities. However, one key limitation of 3D chemical descriptors in QSAR analysis is the computational complexity of conformer generation and structure alignments, which are absent of any guarantees that predicted conformations correspond to relevant bioactive conformations.
Four-dimensional (4D) descriptors are an extension of 3D chemical descriptors that simultaneously consider multiple structural conformations. Ash and Fourches (Ash & Fourches, 2017) applied molecular dynamics simulation on ERK2 kinase to compute 3D descriptors over a grid box based on the 20 ns trajectory and showed that such 4D chemical descriptors can effectively differentiate the most active ERK2 inhibitors from the inactive ones with superior enrichment rates. The ability to consider multiple conformations enhances the accuracy of predictions and allows for a better understanding of the molecular dynamics underlying chemical activity.

2.1.1 3D geometry

A 3D chemical descriptor is a type of molecular representation that characterizes the 3-dimensional structure of a molecule. It provides information about the spatial arrangement of atoms and the distance between them in a molecule.

There are several types of 3D chemical descriptors, including:

Shape descriptors: Shape descriptors are one of the most commonly used 3D chemical descriptors and are used to quantitatively describe the shape of a molecule. This descriptor provides information about the molecular shape that cannot be obtained from other types of descriptors, such as 2D representations or electrostatic potential descriptors. The shape descriptors can be used to compare the shapes

of different molecules, identify molecules that have similar shapes, and analyze the relationship between molecular shape and biological activity. By comparing the shapes of different molecules, researchers can gain insights into the behavior of molecules in different chemical reactions and biological processes. For example, shape descriptors have been used in the design of drugs that have a similar shape to a natural substrate or inhibitor, allowing the drug to bind to the same receptor site and exert a similar effect. Shape descriptors are typically calculated using computational methods, such as molecular mechanics or quantum mechanics calculations. These calculations allow researchers to generate a 3D model of a molecule and calculate descriptors such as the volume, surface area, and curvature of the molecule. The resulting data can then be used to generate a quantitative measure of the shape of the molecule, such as the molecular shape index, surface roughness index, or shape circularity index.

Electrostatic descriptors: Electrostatic descriptors are another type of 3D chemical descriptor that provides information about the electrostatic potential of a molecule. These descriptors are used to predict the reactivity and biological activity of a molecule based on its electrostatic properties. The electrostatic potential of a molecule is determined by the distribution of charged particles, such as electrons and protons, within the molecule. Electrostatic descriptors quantify the electrostatic potential at various points within the molecule, allowing researchers to identify regions of high or low electrostatic potential. These descriptors are typically calculated using computational methods, such as quantum mechanics or molecular dynamics simulations. One commonly used electrostatic descriptor is the molecular electrostatic potential (MEP), which is calculated by solving the Poisson equation for the molecule. The MEP provides a three-dimensional map of the electrostatic potential of the molecule, which can be visualized and analyzed to gain insights into the reactivity and biological activity of the molecule. Electrostatic descriptors have many applications in chemistry and biochemistry. For example, they can be used to predict the binding affinity of a molecule to a receptor site, based on the electrostatic complementarity between the two molecules. They can also be used to predict the reactivity of a molecule in a chemical reaction, based on the electrophilic or nucleophilic character of the molecule. Additionally, electrostatic descriptors can be used to analyze the behavior of molecules in an aqueous environment, as the electrostatic properties of the molecule can influence its solubility and transport properties.

Hydrophobicity descriptors: Hydrophobicity descriptors are another type of 3D chemical descriptor used to calculate the hydrophobicity of a molecule. Hydrophobicity refers to the tendency of a molecule to be insoluble in water or other polar solvents, due to its nonpolar nature. Hydrophobicity descriptors provide information about the distribution of hydrophobic and hydrophilic regions within a molecule, allowing researchers to predict its solubility in different solvents. The hydrophobicity of a molecule can affect its behavior in biological systems, as hydrophobic molecules tend

to accumulate in lipid membranes and can affect membrane function. One commonly used hydrophobicity descriptor is the octanol-water partition coefficient (LogP), which is a measure of the ratio of a molecule's concentration in octanol (a nonpolar solvent) to its concentration in water. A high LogP value indicates that the molecule is more hydrophobic and less soluble in water, while a low LogP value indicates that the molecule is more hydrophilic and more soluble in water. Other hydrophobicity descriptors include the surface area of the molecule that is accessible to water, as well as various measures of the molecule's polarity, such as the polar surface area or the dipole moment. Hydrophobicity descriptors are useful in drug discovery, as the solubility of a drug in different solvents can affect its bioavailability and pharmacokinetics. By using hydrophobicity descriptors to predict a drug's solubility in different solvents, researchers can optimize its formulation to improve its delivery and efficacy.

2.1.2 SMILES representation

The Simplified molecular-input line-entry system (SMILES) is a widely-used notation system for representing the structure of molecules in a compact and easily interpretable format. SMILES notations are strings of characters that represent the atoms and bonds in a molecule, with each atom being represented by its elemental symbol and each bond being represented by a variety of symbols such as '-', '=', and '#'. The use of SMILES notation is widespread in the field of chemistry due to its ability to represent complex structures in a simple, standardized format that can be easily read and interpreted by computers and other software programs. SMILES notations are commonly used for a variety of applications, including molecular modeling, database searching, and predicting the properties of new compounds. One of the key advantages of SMILES notation is its conciseness. The SMILES notation for a given molecule is typically much shorter than other representations, such as the full chemical formula or a 3D molecular structure. This makes it easier to store and manipulate large numbers of molecular structures in computer databases, which are often used for drug discovery and other chemical research applications. In addition to its conciseness, SMILES notation is also highly standardized. This means that SMILES notations are consistent across different software programs and databases, making it easy to share and exchange molecular structures between different users and systems. The standardization of SMILES notation also helps to reduce errors and inconsistencies that can arise when different users or software programs use different notations for the same molecule. Another advantage of SMILES notation is its ability to represent complex chemical structures in a way that is easily interpretable by humans. For example, SMILES notations can represent stereochemistry, which refers to the spatial arrangement of atoms in a molecule. Stereochemistry is an important aspect of many chemical reactions and can have a significant impact on the properties of a molecule.

By representing stereochemistry in a concise and standardized format, SMILES notation makes it easier for researchers to understand and interpret the structures of complex molecules. The SMILES is a highly useful and widely used notation system for representing the structure of molecules. Its conciseness, standardization, and ability to represent complex structures make it a valuable tool for researchers in a variety of fields, from drug discovery to materials science to environmental chemistry.

2.2 Deep learning in molecular design

In this section, we review the three most commonly used machine learning models in the molecular design field.

2.2.1 Recurrent neural networks

Benefiting from the sequential representation of chemical structures, recurrent neural networks (RNNs) (Rumelhart et al., 1986) became a commonly used model for molecular design. It also works as a critical building block of Autoencoders (Baldi, 2012) and Generative adversarial networks (Goodfellow et al., 2020) which will be discussed hereafter. A Recurrent Neural Network (RNN) is a type of artificial neural network that is designed to process sequential data or time series data. This deep learning algorithm is particularly useful for ordinal or temporal problems, such as language translation, natural language processing (NLP) (Nadkarni et al., 2011), speech recognition (Hinton et al., 2012), and image captioning (Xu et al., 2015), and is incorporated into various popular applications, including Siri (Capes et al., 2017), voice search (Hinton et al., 2012), and Google Translate (Wu et al., 2016). Unlike feedforward and convolutional neural networks (CNNs) (LeCun et al., 1998), RNNs have a "memory" that enables them to use information from previous inputs to influence the current input and output. Unlike traditional deep neural networks, which assume that inputs and outputs are independent of each other, RNN outputs depend on the prior elements within the sequence. However, unidirectional RNNs are limited in that they cannot account for future events in their predictions. To illustrate the functioning of RNNs, we can consider an idiom, such as "I am eating noodles," which requires the specific order of words to make sense. RNNs take into account the position of each word in the idiom and use this information to predict the next word in the sequence. RNNs share parameters across each layer of the network, in contrast to feedforward networks, which have different weights across each node. The weight parameter within each layer of the network is adjusted through backpropagation and gradient descent to facilitate reinforcement learning. To determine the gradients, RNNs use the backpropagation through time (BPTT)

(Werbos, 1990) algorithm, which is specific to sequence data. BPTT operates on the same principles as traditional backpropagation, where the model trains itself by calculating errors from its output layer to its input layer. This calculation enables the adjustment and fitting of the model's parameters. However, BPTT differs from the traditional approach by summing errors at each time step, whereas feedforward networks do not need to sum errors since they do not share parameters across each layer. During the training process, RNNs face two common problems: exploding gradients and vanishing gradients. The size of the gradient, which is the slope of the loss function along the error curve, defines these issues. When the gradient is too small, it continues to become smaller, eventually updating the weight parameters to zero, resulting in an algorithm that is no longer learning. Conversely, exploding gradients occur when the gradient is too large, leading to an unstable model. In this case, the model weights will grow too large, eventually becoming represented as NaN. One solution to these issues is to reduce the number of hidden layers in the neural network, thereby reducing the complexity of the RNN model.

2.2.2 Autoencoders

In the field of machine learning, autoencoders have become an increasingly popular technique for unsupervised learning. Autoencoders leverage neural networks to learn and represent the underlying structure of data by imposing a bottleneck in the network, which forces a compressed knowledge representation of the original input. The resulting compressed representation can be used for various tasks, such as data compression, anomaly detection (Sakurada & Yairi, 2014), and feature extraction (Masci et al., 2011). This essay will discuss the working principles of autoencoders, including the design of the neural network architecture, and the significance of the imposed bottleneck. It will also explore the relationship between input feature independence and the difficulty of the reconstruction task. Autoencoders are composed of an encoder network and a decoder network. The encoder network takes the input data and compresses it into a lower dimensional representation, while the decoder network takes the compressed representation and attempts to reconstruct the original input. The key feature of the autoencoder architecture is the bottleneck layer which forces the encoder to learn a compressed representation of the input data. This bottleneck is often a hidden layer with fewer nodes than the input layer, forcing the encoder to learn a compressed representation of the input data. The bottleneck layer's purpose is to force the encoder to learn a compressed representation of the input data. This compressed representation is often referred to as the latent space or code. The size of the bottleneck layer determines the amount of compression applied to the input data, and the quality of the reconstruction. If the bottleneck layer is too small, the network may not be able to learn a useful representation of the input data. On the other

hand, if the bottleneck layer is too large, the network may not be able to capture the relevant structure of the input data and produce a low-quality reconstruction. The success of an autoencoder relies heavily on the existence of correlations between the input features. If the input features are independent of one another, the task of compression and reconstruction becomes extremely difficult. However, if there is some structure or correlation between input features, autoencoders can learn this structure and use it to compress and reconstruct the input data. For example, in image processing, neighboring pixels are often correlated, and autoencoders can learn this correlation to compress and reconstruct the image. In contrast, if the input data lacks any structure, autoencoders cannot learn any useful representation. Autoencoders have a wide range of applications, ranging from data compression to feature extraction. In data compression, autoencoders can be used to compress large datasets to reduce storage and computational requirements. Autoencoders can also be used for anomaly detection, where they can detect anomalous data points by comparing the reconstruction error to the original input. In addition, autoencoders can be used for feature extraction, where they learn a compressed representation of the input data, which can be used as input for supervised learning algorithms.

Autoencoders have emerged as a promising technique in the field of molecular design due to their ability to generate new molecules by making small modifications to the latent feature space. This is achieved through the use of a decoder network that reconstructs the original input from the compressed latent representation, which can also be seen as a generative model. One of the advantages of autoencoders in molecular design is that they can learn the underlying structure and patterns in the molecular data, which allows for the generation of new molecules with desirable properties. The decoder network can be trained to produce new molecules by modifying the latent feature space, which can lead to slightly different chemical structures compared to the original input. This process can be used to explore the chemical space and identify novel molecules that may have beneficial properties. The generation of new molecules using autoencoders has been shown to be an effective approach for drug discovery and materials design. For example, in drug discovery, the goal is to identify new molecules that have a desired therapeutic effect while minimizing potential side effects. Autoencoders can be trained on a dataset of known drug molecules and then used to generate new molecules with similar properties. These generated molecules can then be evaluated for their potential as drugs, and those that show promise can be further developed and tested. Similarly, in materials design, autoencoders can be used to generate new materials with specific properties, such as conductivity, strength, or flexibility. By modifying the latent feature space, the decoder network can generate new materials that have slightly different properties compared to the input data. These new materials can be evaluated and selected for their potential use in various applications. Overall, the ability of autoencoders to generate new molecules

by modifying the latent feature space has made them a valuable tool in molecular design. By leveraging the underlying structure and patterns in the molecular data, autoencoders can be used to explore the chemical space and identify novel molecules with desirable properties. As such, they have the potential to revolutionize drug discovery and materials design, and their use in these fields is expected to continue to grow in the future.

2.2.3 Generative adversarial networks

Generative Adversarial Networks (GANs) (Goodfellow et al., 2020) are a machine learning model that has gained significant attention in recent years. GANs employ deep learning methods to generate new data from an existing dataset by using two competing neural networks, known as the generator and the discriminator, which work together in a cooperative zero-sum game framework. The generator is trained to produce synthetic data, while the discriminator is trained to differentiate between the synthetic and real data. The generator is typically a convolutional neural network (CNN), and the discriminator is a deconvolutional neural network (DNN) (Zeiler & Fergus, 2014). The goal of the generator is to produce data that is so realistic that it cannot be distinguished from the real data by the discriminator. On the other hand, the discriminator aims to distinguish between the synthetic data and the real data. One of the key features of GANs is their ability to generate new data by creating a probabilistic model. This means that GANs can learn the underlying structure of the dataset and use it to create new data that closely resembles the original. For example, a GAN can generate realistic-looking images of human faces that do not correspond to any real individual. The process of training a GAN involves several steps. Firstly, an initial training dataset is collected based on the desired output. This dataset is then randomized and fed into the generator to produce synthetic data. The synthetic data is combined with the real data and inputted into the discriminator. The discriminator then calculates a probability score between 0 and 1 to determine whether the input data is real or synthetic. If the discriminator correctly identifies the synthetic data, a penalty is applied to the generator, and the process is repeated until the generator produces data that is indistinguishable from the real data. The training process of GANs can be divided into three main categories: generative, adversarial, and networks. The generative category focuses on how data is generated based on a probabilistic model, while the adversarial category focuses on the competitive aspect of the model. The network category refers to the use of deep neural networks as the artificial intelligence algorithms for training purposes.

2.3 Design objective

In machine learning, choosing the appropriate evaluation metric is critical to obtain meaningful insights into the performance of a model in a given application. Evaluation metrics provide a quantitative measure of how well a machine learning model is able to perform a specific task, in contrast to training methods that aim to improve the model's performance during training. Based on the reference of the metric, evaluation metrics can be categorized into three groups. In the field of molecular generation, it is common practice to use a set of evaluation metrics to assess the quality of generated molecules. One of the most important groups of evaluation metrics is the one that involves comparing the generated molecules with the training data. This group of metrics is typically employed when the goal is to reconstruct existing data. Within this group of evaluation metrics, there are several specific criteria that researchers can use to determine the accuracy of the model's ability to replicate known molecules. One such metric is the coverage of existing data, which measures how well the generated molecules represent the entire training set. This metric is crucial because it assesses how well the model has learned to represent the diversity of the training data. Another metric in this group is the distance to the existing distribution, which measures how similar the generated molecules are to the molecules in the training set. This metric is important because it evaluates the model's ability to generate molecules that are consistent with the underlying distribution of the training data. By analyzing this metric, researchers can determine how well the model has learned to capture the underlying patterns and structures of the training data. The evaluation of the similarity between the generated molecules and the training data is a critical step in the assessment of the quality of molecular generation models. By using a range of evaluation metrics, researchers can gain a more comprehensive understanding of the model's strengths and weaknesses, and ultimately improve the accuracy and reliability of molecular generation techniques.

The second group of evaluation metrics is critical for assessing the model's ability to design molecules that exhibit specific properties. The machine learning model is tasked with generating molecules that meet specific design criteria. This process involves optimizing the chemical structure of the molecule to achieve the desired properties. To determine the effectiveness of the model's ability to generate molecules with specific properties, there are several evaluation metrics that researchers use. One of the most common metrics is measuring the desired property of the molecule, such as potency, selectivity, or toxicity. These properties are essential in drug design, where the goal is to create a molecule with specific biological activity that can interact with a target protein or enzyme. In addition to measuring the desired properties of the molecule, researchers may also evaluate other properties such as solubility, stability, and bioavailability. These properties are crucial in determining whether

a molecule can be developed into a viable drug candidate. Overall, evaluating the properties of the designed molecules is a critical step in the assessment of the quality of machine learning models in molecular design. By using a range of evaluation metrics, researchers can assess the model's ability to generate molecules that meet specific design criteria and identify areas for improvement. This can lead to the development of more accurate and effective molecular design techniques, which have the potential to revolutionize the field of drug discovery and other related areas.

The last group of evaluation metrics involves assessing the quality of the molecules generated by the machine learning model in molecular design. The quality of the generated molecules is a critical aspect of the evaluation process because it determines the usefulness and practicality of the model's outputs. In this group of evaluation metrics, researchers assess the feasibility, stability, and synthetic accessibility of the generated molecules. Feasibility refers to the practicality of synthesizing molecules in a laboratory setting. Stability is a measure of the molecule's ability to maintain its chemical structure under different conditions. Synthetic accessibility is a measure of the ease with which the molecule can be synthesized using existing chemical methods. To assess the feasibility of the generated molecules, researchers may use computational tools to predict the synthetic pathway of the molecule and its likelihood of being synthesized in a laboratory. The stability of the generated molecules can be assessed using various computational techniques, including molecular dynamics simulations and quantum chemical calculations. Synthetic accessibility can be evaluated using computational tools that predict the ease of synthesizing the molecule using existing chemical methods. By evaluating the quality of the generated molecules, researchers can determine if the machine learning model is generating realistic and chemically feasible molecules. If the model is generating molecules that are difficult or impossible to synthesize, then the model's practical utility is limited. On the other hand, if the model generates high-quality molecules that are easy to synthesize and have desirable properties, then it has the potential to be a valuable tool in drug discovery and other molecular design applications. The last group of evaluation metrics is an essential component of assessing the quality of machine learning models in molecular design. By evaluating the feasibility, stability, and synthetic accessibility of the generated molecules, researchers can determine the practicality and usefulness of the model's outputs.

Chapter 3

Bayesian inference in molecular design

In the previous chapter, we did a survey about the general building blocks of machine learning-based molecular design. Other than this, some specific methods and algorithms are involved in this work. In this chapter, we will go through the details of this stuff.

3.1 Forward model: synthetic reaction

Suppose that a single-step reaction is given as

$$S_1 + S_2 \rightarrow P. \tag{3.1}$$

S_1 and S_2 denote two reactants, and P denotes the product, which is assumed to be a singleton as byproducts are ignored here. In this study, we considered only synthetic reactions with two reactants. In this study, we considered only synthetic reactions with two reactants, but the proposed method can be generalized to handle any number of reactants. Our software, which will be introduced later, can design reaction pathways without limiting the number of reactants. In addition, solvents and reagents can also be incorporated on demand.

Consider that the single-step reaction is modeled by a function r as

$$P = r(S), \tag{3.2}$$

where $S = \{S_1, S_2\}$. Function r represents the change in the chemical structure from S to P. Note that r is a set function that is invariant to the exchange of S_1 and S_2. As described later, the single-step reaction prediction model was modeled using Molecular Transformer.

Arbitrary reaction networks can be modeled by combining the single-step model. For example, consider a three-step single-chain reaction as

$$\begin{aligned} &\text{Step 1} : S_{11} + S_{12} \to P_1 \\ &\text{Step 2} : P_1 + S_{21} \to P_2 \\ &\text{Step 3} : P_2 + S_{31} \to P_3 \ (=: P). \end{aligned} \tag{3.3}$$

In the first step, two reactants, S_{11} and S_{12}, produce the intermediate product P_1, followed by the second step, which produces the second intermediate product P_2 by reacting P_1 and a newly selected reactant S_{21}. In the third step, the intermediate product P_2 and reactant S_{31} react to produce final product $P := P_3$. This reaction cascade can be expressed using the single-step reaction model, as follows:

$$\begin{aligned} P &= r(P_2, S_{31}) \\ &= r(r(P_1, S_{21}), S_{31}) \\ &= r(r(r(S_{11}, S_{12}), S_{21}), S_{31}) \\ &=: g(S|G). \end{aligned} \tag{3.4}$$

The final product $P = P_3$ is described as a function $g(\cdot|G)$ of the four purchasable reactants $S = \{S_{11}, S_{12}, S_{21}, S_{31}\}$, where all the intermediate reaction states are discarded. The structure of the reaction network is represented by the synthetic graph or network G. The graph forms a rooted tree in which the leaf nodes consist of the four reactants in S and the root node is given by the final product. Every node, except the leaf nodes, has two children. Without loss of generality, any synthetic reaction network, beyond single-chain reactions, can be described as $P = g(S|G)$ and retains the graph properties. Several examples are shown in Figure 3.1.

3.2 Forward model: property prediction models and scoring function

Once the final product P is generated as $P = g(S|G)$ with any given S, we estimate its properties Y using the prediction model $Y = h(P)$ as

$$\begin{aligned} Y &= h(P) \\ &= h \circ g(S|G) \\ &=: f(S|G). \end{aligned} \tag{3.5}$$

As the product can be represented by the deterministic function $g(S|G)$ of S, Y can be expressed by a function of S as $Y = f(S|G)$. Here, Y can be a vector of one

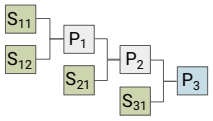
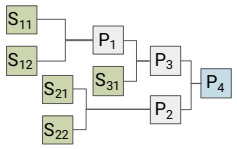
Network type	Reaction sequence	Reaction prediction model	Graph representation
Single-step	$S_1 + S_2 \to P$	$P = r(S_1, S_2)$	
Three-step single-chain	$S_{11} + S_{12} \to P_1$ $P_1 + S_{21} \to P_2$ $P_2 + S_{31} \to P_3$	$P_3 = r(P_2, S_{31})$ $= r(r(P_1, S_{21}), S_{31})$ $= r(r(r(S_{11}, S_{12}), S_{21}), S_{31})$	
Four steps (Branches)	$S_{11} + S_{12} \to P_1$ $S_{21} + S_{22} \to P_2$ $P_1 + S_{31} \to P_3$ $P_2 + P_3 \to P_4$	$P_4 = r(P_2, P_3)$ $= r(r(S_{21}, S_{22}), r(P_1, S_{31}))$ $= r(r(S_{21}, S_{22}), r(r(S_{11}, S_{12}), S_{31}))$	

Figure 3.1: Examples of synthetic reaction networks.

or more properties. Function h can be specified arbitrary; for example, a machine learning property prediction model or a scoring function such as the quantitative estimate of drug-likeness (QED) score (Bickerton et al., 2012).

The objective of molecular design is to identify a reactant set S with the resulting P exhibiting a set of desired properties Y^* with respect to the given forward model $Y = f(S|G)$. For the molecular design task, it is necessary to define a measure of the discrepancy $d(Y, Y^*)$ between the predicted properties and the target. A typical example of a discrepancy measure is the Euclidean distance:

$$d(Y, Y^*) = \|Y - Y^*\|^2. \tag{3.6}$$

Additionally, various measures can be defined depending on the type of task. For example, if the target property is given as a region U^*, we can use the following 0-1 loss:

$$d(Y, U^*) = \begin{cases} 0 & Y \in U^* \\ 1 & \text{otherwise} \end{cases}. \tag{3.7}$$

Hereafter, we consider $d(Y, U^*)$, but we can use any type of discrepancy without loss of generality. Alternatively, in the case of a score-type monotonic measure such as QED, d can be defined, for example, as

$$d(Y) = -\text{QED}(Y). \tag{3.8}$$

Furthermore, in the task of retrosynthetic prediction, where a target product P^* is given as the design purpose, we can use, for example, the 0-1 loss between P and P^* as follows:

$$d(P, P^*) = \begin{cases} 0 & P = P^* \\ 1 & \text{otherwise} \end{cases}. \tag{3.9}$$

This is equivalent to setting $h(P) = 1$ and $Y = P$ in the property prediction model, Eq. 3.5.

3.3 Bayesian inverse problem

Herein, we describe the task of designing molecules with their synthetic routes. Suppose that a collection of N commercially available compounds is given by

$$\mathcal{B} = \{S_1, ..., S_N\}. \tag{3.10}$$

A reactant set S that forms the designed G should be selected from $\mathcal{B}$. Here, by $\mathcal{P}_k(\mathcal{B})$, we denote the set of all k combinations from $\mathcal{B}$. Then, the support $\mathcal{P}(\mathcal{B})$ of S is expressed as

$$\mathcal{P}(\mathcal{B}) = \mathcal{P}_1(\mathcal{B}) + \cdots + \mathcal{P}_K(\mathcal{B}), \tag{3.11}$$

where the maximum number of reactants that can be selected is constrained to K.

The design variable is denoted by a tuple $x = \{S, G\}$. Here, we write the model in Eq. 3.5 as $Y(x) = f(x) = f(S|G)$ in order to explicitly express that the predicted property Y is a deterministic function of the reactant S and the reaction network G. As in our previous studies (Ikebata et al., 2017; Guo et al., 2020), molecular design based on Bayesian inference is performed based on the target distribution $\pi(x)$, which is defined on $\mathcal{P}(\mathcal{B})$:

$$\begin{aligned} \pi(x) &= p(x|Y \in U^*) \\ &\propto p(x, Y \in U^*) \\ &= p(Y \in U^*|x)p(x) \\ &= \frac{1}{Z} \exp\left(-\frac{1}{\sigma} d(Y(x), U^*)\right) \cdot p(x). \end{aligned} \tag{3.12}$$

According to Bayes' law, the posterior distribution $\pi(x) = p(x|Y \in U^*)$ is proportional to the joint distribution $p(x, Y \in U^*)$, which consists of the product of the likelihood $p(Y \in U^*|x)$ and prior distribution $p(x)$. The forward model

forms the joint probability distribution, which is modeled by the Gibbs distribution $p(Y \in U^*|x) = \frac{1}{Z}\exp\left(-\frac{1}{\sigma}d(Y(x), \mathbf{U}^*)\right)$, with temperature parameter $\sigma > 0$. The normalizing constant Z is the canonical partition function:

$$\begin{aligned} Z &= \sum_{Y \in U^*, Y \notin U^*} \exp\left(-\frac{1}{\sigma}d(Y(x), U^*)\right), \\ &. = \exp\left(-\frac{1}{\sigma} \times 0\right) + \exp\left(-\frac{1}{\sigma} \times 1\right) \\ &= 1 + \exp\left(-\frac{1}{\sigma}\right) \end{aligned} \tag{3.13}$$

The prior distribution $p(x)$ is used to narrow down the broad solution space based on prior knowledge. For example, the prior can be modeled as follows:

$$p(x) \propto I(S \subset \mathcal{P}(\mathcal{B})) \cdot \exp\left(-\frac{|S|}{\tau}\right) \cdot \exp\left(-\frac{|G|}{\tau'}\right). \tag{3.14}$$

The indicator function $I(\cdot)$ takes the value one or zero depending on whether the designed reactant set S belongs to a subset of purchasable compounds. The second and third terms on the right-hand side penalize the increasing number of reactants ($|S|$) and the size of the designed network ($|G|$), where τ and τ' determine the magnitude of penalties.

By identifying x with a sufficiently high posterior probability, we predict product P and its synthetic reaction route $\{S, G\}$ that satisfy design objective U^*. However, computational difficulty here arises from the extremely large search space. The search space is composed of all combinations of reactants that are purchasable. The number of candidate reactants is typically of the order $O(10^6)$, resulting in the cardinality of the solution space burgeoning to $O(10^{6\times k})$ when k reactants are involved in the synthetic route planning. In addition, the network topology to be explored further increases the size of the search space.

3.4 Sequential Monte Carlo in general

For the main building block of the proposed method, we employed a sequential Monte Carlo (SMC) algorithm (Del Moral et al., 2006) to draw a promising sample set of $x = \{S, G\}$ from the target $\pi(x)$. Basically, most of the heuristic algorithms can be incorporated into the proposed method. However, as will be introduced later, the proposed method uses a real-time procedure to design the synthetic route and SMC is designed for real-time learning and estimation in dynamic and sequential systems. This is due to several key characteristics of SMC methods that make them well-suited

for applications that involve updating estimates as new data becomes available. So we employ SMC in this study. As mentioned above, because the target distribution is defined on a large combinatorial space, an ordinary SMC cannot approximate it adequately. In particular, it is difficult to obtain a diverse set of highly probable molecules with their reaction routes using such ordinary methods, which will be demonstrated later. The proposed method, shown in the next section, was developed as an extension of SMC to overcome the difficulty of combinatorial complexity. To clarify the design concept of the proposed method, we briefly describe the general SMC method here.

In conventional SMC, we define an augmented target distribution $\pi_A(\boldsymbol{x})$ using T auxiliary distributions, $\pi_t(x_t)$ $(t = 1, \ldots, T)$, and an arbitrarily chosen initial distribution $\pi_0(x_0)$:

$$\pi_A(\boldsymbol{x}) = \pi_0(x_0) \prod_{t=1}^{T} \pi_t(x_t). \tag{3.15}$$

The augmented variable $\boldsymbol{x} = (x_0, x_1, \ldots, x_T)$ consists of $T + 1$ auxiliary variables. The goal of SMC is to efficiently approximate the entire system $\pi_A(\boldsymbol{x})$ with the Monte Carlo approximator, successively sampling x_t in the order $t = 0, 1, \ldots, T$. The definition of the auxiliary distribution is arbitrary. For example, if the last auxiliary distribution is defined as the original target $\pi_T(x_T) = \pi(x)$, we can take a sample set of x_T to obtain an approximate distribution. Alternatively, all T auxiliary distributions $\pi_1(x_1) \ldots, \pi_T(x_T)$ can be set to be identical to the target $\pi(x)$. In this case, all samples of $x_1, \ldots, x_T$ can be used for the approximate inference. The essence of SMC methodology is to be able to use any sequence of auxiliary distributions to efficiently obtain random samples from the intractable target distribution. As a constraint to be satisfied, it is imposed that the last auxiliary distribution is consistent with the target distribution.

To derive the sampling algorithm, we rewrite the augmented distribution in Eq. 3.15 as

$$\pi_A(\boldsymbol{x}) = \pi_0(x_0) \prod_{t=1}^{T} \left\{ \frac{\pi_t(x_t)}{\eta(x_t|x_{t-1})\pi_{t-1}(x_{t-1})} \right\} \eta(x_t|x_{t-1})\pi_{t-1}(x_{t-1}). \tag{3.16}$$

The conditional probability distribution $\eta(x_t|x_{t-1})$, called the proposal distribution, determines the transition process from x_{t-1} to x_t. Assume that we currently have a sample set $\{x_{t-1}^i | i = 1, \ldots, m\}$ of x_{t-1} that follows $\pi_{t-1}(x_{t-1})$. This set defines a Monte Carlo approximation of $\pi_{t-1}(x_{t-1})$ as

$$\hat{\pi}_{t-1}(x_{t-1}) = \frac{1}{m} \sum_{i=1}^{m} I(x_{t-1} = x_{t-1}^i) \tag{3.17}$$

The indicator function $I(\cdot)$ takes the value one if the argument is true; otherwise, it takes zero. The purpose of each step of SMC is to derive a conversion from $\hat{\pi}_{t-1}(x_{t-1})$ to $\hat{\pi}_t(x_t)$ based on the form in Eq. 3.16. To be specific, consider the following recursive formula derived by substituting the approximate distribution $\hat{\pi}_{t-1}(x_{t-1})$ into Eq. 3.16:

$$\begin{aligned}\hat{\pi}_t(x_t) &= \left\{ \frac{\pi_t(x_t)}{\eta(x_t|x_{t-1})\pi_{t-1}(x_{t-1})} \right\} \eta(x_t|x_{t-1})\hat{\pi}_{t-1}(x_{t-1}) \\ &= \frac{1}{m}\sum_{i=1}^{m} w(x_t|x_{t-1}^i)\eta(x_t|x_{t-1}^i) \quad (t = 1, \ldots, T).\end{aligned} \tag{3.18}$$

Based on this form with the given $\hat{\pi}_{t-1}(x_{t-1})$, we generate a sample set $\{x_t^i|i = 1, \ldots, m\}$ of x_t from $\hat{\pi}_t(x_t)$ by performing the sampling importance resampling (SIR) method (Rubin, 1988) with the importance weight given by $w(x_t|x_{t-1}^i) = \frac{\pi_t(x_t)}{\eta(x_t|x_{t-1}^i)\pi_{t-1}(x_{t-1}^i)}$. The algorithm starts with an initial sample set $\{x_0^i|i = 1, \ldots, m\} \sim \pi_0(x_0)$ and repeats the following steps for $t = 1, \ldots, T$:

(1) Draw a particle x_t^i from the proposal distribution $\eta(x_t|x_{t-1}^i)$ for each of $i = 1, \ldots, m$.

(2) Calculate the importance weight $w(x_t^i|x_{t-1}^i)$ to obtain the approximate distribution as

$$\hat{\pi}_t(x_t) = \frac{1}{\sum_{i=1}^{m} w(x_t^i|x_{t-1}^i)} \sum_{i=1}^{m} w(x_t^i|x_{t-1}^i) I(x_t = x_t^i). \tag{3.19}$$

(3) Resample $\{x_t^i|i = 1, \ldots, m\}$ with probability proportional to $w(x_t^i)$ to renew the m samples as following an empirical distribution $\hat{\pi}_t(x_t) = \frac{1}{m}\sum_{i=1}^{m} I(x_t = x_t^i)$ with the equal weight as in Eq. 3.17.

In step 1, the previous sample set is tentatively replaced with a new one according to the proposal distribution $\eta(x_t|x_{t-1}^i)$. The proposal distribution is designed to search a neighboring area of the currently obtained x_{t-1}^i with a certain probability and to search a still unexplored region with the remaining probability. This strikes a balance between exploitation and exploration. In step 2, we calculate the importance weight representing the goodness-of-fit of the proposed x_t^i. Finally, in step 3, we determine the survival or death of x_t^i according to the importance weight. The SMC is essentially equivalent to a genetic algorithm.

As shown later, the simple SMC performs very poorly in our molecular design task. The support of the posterior distribution is extremely large; moreover, the promising solution sets are widely scattered in the huge support. Because conventional SMC cannot cope with the target task, we developed an extended method following the SMC framework.

Chapter 4

Proposal

4.1 Recurrent design algorithm for synthetic reaction networks

Here, we begin by depicting the key idea of the recurrent design technique for reaction networks following the example shown in Figure 4.1. Note that for any reaction network, the final product at the root node has two reactants in its parent nodes. We consider that at each step of SMC, denoted by t, only the two reactants of the final product are explored. In the forward cascade model $Y = h \circ g(S|G)$, only the single-step reaction $S_1 + S_2 \rightarrow P$ is considered as G, and the two reactants $\{S_1, S_2\}$ are selected from the original set $\mathcal{B}$ of commercial compounds and an additional set $\mathcal{I}_{t-1}$ containing all intermediate products computationally synthesized until step $t-1$. Let $\mathcal{B}_t = \mathcal{B} \cup \mathcal{I}_{t-1}$ be the expanded set of candidate reactants and let $x_t = \{S_1, S_2\}$ be the auxiliary variable at t. Then, the auxiliary distribution $\pi_t(x_t)$ is defined as

$$\pi_t(x_t) \propto \exp\left(-\frac{1}{\sigma} d(Y(x_t), U^*)\right) \text{ where } \forall x_t \in \mathcal{B}_t \times \mathcal{B}_t : \; Y(x_t) = h \circ r(S). \quad (4.1)$$

Unlike the general form in Eq. 3.5, the forward model $Y(x_t) = h \circ r(S)$ represents only the single-step reaction model $r(S)$, and consequently the auxiliary distribution $\pi_t(x_t)$ is a function of two reactants only. Nevertheless, the model is able to represent general reaction networks by selecting the already calculated intermediate products contained in $\mathcal{B}_t$, to which their reaction networks are implicitly assigned.

Specifically, we proceed with the following SMC procedure. In step $t-1$, a sample set $\{x_{t-1}^i | i = 1, \ldots, m\}$ of reactant set x_{t-1} is obtained, and a new set $\{x_t^i | i = 1, \ldots, m\}$ is generated using the proposal $\eta(x_t|x_{t-1})$, where each x_t is sampled from $\mathcal{B}_t$. Here, all newly generated products $P_i = r(x_t^i)$ $(i = 1, \ldots, m)$ are added to the set of candidate

reactants in the next step, as follows:

$$\mathcal{B}_{t+1} = \mathcal{B}_t \cup \{P_1, \ldots, P_m\}. \tag{4.2}$$

Thus, the support $\mathcal{B}_t$ of the auxiliary distribution constitutes a sequence of increasing sets as

$$\mathcal{B} = \mathcal{B}_0 \subseteq \mathcal{B}_1 \subseteq \cdots \subseteq \mathcal{B}_{T-1} \subseteq \mathcal{B}_T. \tag{4.3}$$

At each step, the reactants are sampled from the auxiliary distribution $\pi_t(x_t)$ defined on $\mathcal{B}_t \times \mathcal{B}_t$ following the procedure described above. After calculating the predicted products $P_i = r(x_t^i)$ using the single-step reaction model and predicted properties $Y_i = h \circ r(x_t^i)$, and calculating the importance weights, resampling is performed to determine the survival or death of x_t^i. Because the solution space for each t is restricted to the support of two reactants $\mathcal{B}_t \times \mathcal{B}_t$, no combinatorial explosion occurs. Additionally, as t increases, the network topology that can be represented by $\mathcal{B}_t$ increases monotonically. In principle, if t approaches infinity under the use of a proper proposal distribution, then $\mathcal{B}_t$ can represent any network topology. We call this method the Bayesian sequential stacking algorithm, which describes the process of constructing a reaction network by recurrently stacking single-step reactions (Figure 4.1).

For the proposal $\eta(x_t|x_{t-1})$, we employ a mixture model such that a neighboring reactant of each in x_{t-1} is selected from $\mathcal{B}_t$ with probability α, and with probability $1-\alpha$, x_t is chosen completely at random from $\mathcal{B}_t \times \mathcal{B}_t$ in order to obtain a renewed x_t. Probability α is a hyperparameter that controls the trade-off between "exploitation" and "exploration". The "exploration" creates a mechanism that enhances the diversity of solutions. The problem we face here is the computational cost of the neighborhood search. The set of candidate reactants $\mathcal{B}_t$ grows monotonically with each step. Calculating the similarity between all entries in $\{x_{t-1}^i | i = 1, \ldots, m\}$ and the monotonically increasing $\mathcal{B}_t$ at every step is quite time-consuming. Therefore, we introduce a method to reduce the computational complexity, as described below.

The initial set $\mathcal{B}$ of commercial compounds is divided into K clusters according to the pattern of the chemical structures. First, the chemical structure S is transformed into a descriptor vector $\phi(S)$ of length 3239 with the concatenation of RDKit fingerprint (Landrum et al., 2006) (length 2048), MACCS (Molecular ACCess System) keys (Durant et al., 2002) (length 167), and Morgan fingerprint (Morgan, 1965) of radius 2 (length 1024). Then, generative topographic mapping (GTM) is applied (Bishop et al., 1998), which is a well-established integrated method of dimensionality reduction and clustering, in order to obtain a unique mapping from vectorized chemical structures to class labels as

$$k = k(S) \equiv k(\phi(S)) \text{ where } k \in \{1, \ldots, K\}. \tag{4.4}$$

Thus, an arbitrary compound set can be partitioned into K disjoint clusters (Figure 4.2).

Using a trained GTM, candidate reactants that are newly added to $\mathcal{B}_{t-1}$ are sequentially grouped into each of the K predefined clusters $\mathcal{C}_k^{t-1}$ $(k = 1, \ldots, K)$. Here, $R \in \mathcal{C}_k$ denotes an element of cluster k, where the cluster members vary with step t; here, we omit the subscript indicating the dependence of the cluster on t. The model $\eta_0(x_t|x_{t-1})$ for the neighborhood search in the proposal distribution selects $x_t = \{S_{1,t}, S_{2,t}\}$ with equal probability from cluster $R \in \mathcal{C}_{k(S_{i,t-1})}$ $(i = 1, 2)$ to which each reactant $S_{1,t-1}$ or $S_{2,t-1}$ in x_{t-1} belongs. The explicit form of the probabilistic model can be expressed as

$$\eta_0(x_t|x_{t-1}) = \prod_{i=1}^{2} \prod_{k=1}^{K} \left(\frac{1}{|\mathcal{C}_k|} \right)^{I(S_{i,t} \in \mathcal{C}_{k(S_{i,t-1})})} . \tag{4.5}$$

For the model corresponding to "exploration", we sample a candidate from $\mathcal{B}_t$ with equal probability. In summary, the proposal distribution is given by the two-component mixture distribution as

$$\eta(x_t|x_{t-1}) = \alpha\ \eta_0(x_t|x_{t-1}) + (1 - \alpha) \prod_{i=1}^{2} \left(\frac{1}{|\mathcal{B}_t|} \right)^{I(S_{i,t} \in \mathcal{B}_t)} . \tag{4.6}$$

The first and second terms function as the "exploitation" and "exploration" mechanisms, respectively. Note that the number of clusters K in GTP can also be interpreted as a hyperparameter that determines the trade-off between exploitation and exploration in SMC. When the number of clusters is increased, the homogeneity of molecules within a cluster increases. Therefore, the similarity of the candidate molecules generated from the proposal distribution will also increase. This corresponds to placing a higher weight on the exploitation mechanism. Conversely, when the number of clusters is small, the probability of being replaced by a candidate molecule with a large structural difference tends to increase.

4.2 Acceleration techniques

Here, we consider the problem of slow computation of the deep neural network for the reaction prediction. In this study, we used Molecular Transformer developed by Schwaller et al. (2019) for the forward prediction of a single-step reaction. In our test, a single-step reaction prediction of $m = 500$ particles using Molecular Transformer required 25–40 seconds on average on a Linux server with an NVIDIA V100 GPU. When the number of steps was set as $T = 5000$, the total runtime of the reaction prediction exceeded 56 hours.

We therefore used a computationally light surrogate model $Y = d(S)$ that directly predicts property Y from a given set of reactants S without going through the reaction prediction to generate the product. The chemical structure of each reactant in S was converted into a binary vector of length 3239 by concatenating the Morgan fingerprint with radius 2 and bit length 1024, the MACCS keys, and the RDKit fingerprint. The intersection of the binary vectors for the two reactions in S was then taken to obtain a single-descriptor vector. From the United States Patent and Trademark Office (USPTO) open chemical reaction dataset (Lowe, 2017), which contains 1.1M reactions, we selected a subset of 492k reactions that involve exactly two reactants. We then substituted the selected reactant pairs in Molecular Transformer to obtain their products and calculated their model properties Y using the property predictor h. The gradient boosting regressor (Schapire & Freund, 2013) was then trained to learn the mapping from S to Y using 80% of the 492k instances as the training set. For hyperparameter tuning, using 20 candidate values for the learning rate, 15 for the max depth, and 10 for the number of estimators, we performed 10-fold cross-validation looped within the training set and selected the optimized hyperparameters attaining the smallest mean absolute error (MAE). In the calculation of the importance weights in the SMC module, the surrogate $Y(x_t) = d(S)$ was used instead of $Y(x_t) = h \circ r(S)$. Samples drawn from the proposal distribution at each step of the SMC were handed over to the validation module with Molecular Transformer r and property predictor h for the exact calculation of their synthetic products and resulting properties.

We also considered a technique for parallel computing, wherein the entire algorithm was divided into two modules: SMC and the aforementioned validation calculation of products and properties using the forward models. The SMC module sequentially feeds the reactants sampled at each step into the module of the forward model. The forward-prediction module calculates the synthetic products and properties of the received reactants and adds the calculated products to the pool of candidate reactants in the SMC module. These two units carry out the computation of the reactant sets flowing from the different steps of SMC. The two modules perform their tasks simultaneously once the data exchange is complete, synchronizing the timing of the data exchange, as schematically described in Figure 4.3. The SMC module proceeds with its calculation as $t = 0, 1, \ldots$ without any suspension (top row in Figure 4.3). In contrast, the forward-prediction module is subject to an idle state. Once the calculation of one step of the SMC is completed, the set of reactants produced there is handed over to the module for the forward calculation in an idle state. The forward-prediction module then calculates the predicted products using Molecular Transformer, which are sent back to the SMC module. Once the data transfer is completed, the forward-prediction module returns to the idle state and waits to receive the next reactant set. In the example shown in Figure 4.3, we assume that the SMC module is approximately 1.5 times faster than the forward-prediction module. In this

case, three forward-prediction modules are allocated to different processing units in order to reduce the idling time as much as possible.

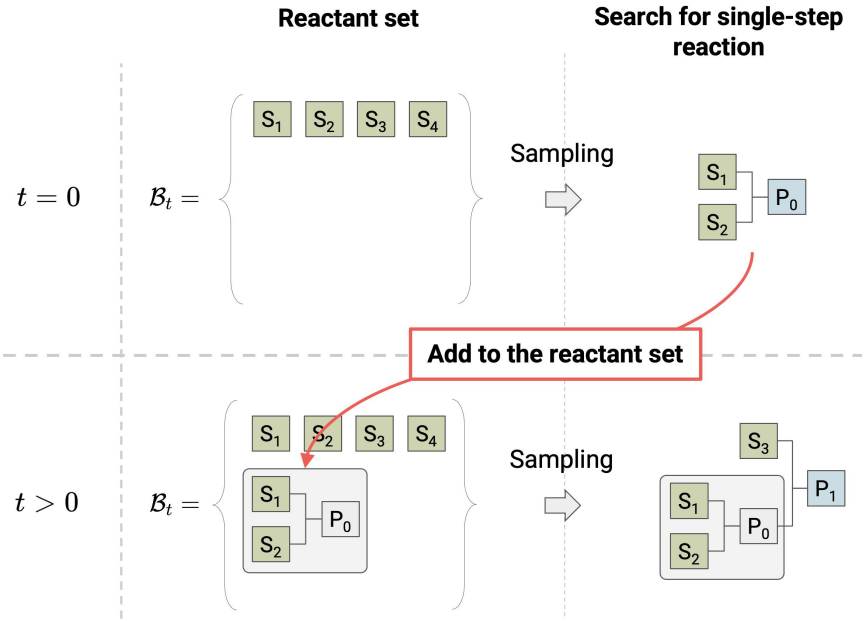


Figure 4.1: Design of synthetic reaction networks using the SMC calculation based on sequential stacking algorithm. In the current step, all sampled reaction networks and their products are added to the reactant pool in the next step. By searching for single-step reactions from this expanding pool of reactants, the network can be built up recursively.

4.3 Summary: Bayesian sequential stacking algorithm for molecular design

The proposed molecular design algorithm is summarized in Algorithm 1. The overall workflow consists of two modules: SMC and calculation of the forward model. The SMC module performs posterior sampling of the single-step reactions using the current pool $\mathcal{B}_t$ of the candidate reactants. The sampling calculation consists of (1) resampling based on the goodness-of-fit of the current set of reactants, (2) updating the set of hypothetical reactants based on the proposed distribution, and (3) calculating the goodness-of-fit importance weights. All reactant sets generated from the proposal distribution are handed over to the forward model module to generate the predicted products and characterize their properties. All the products generated in this module

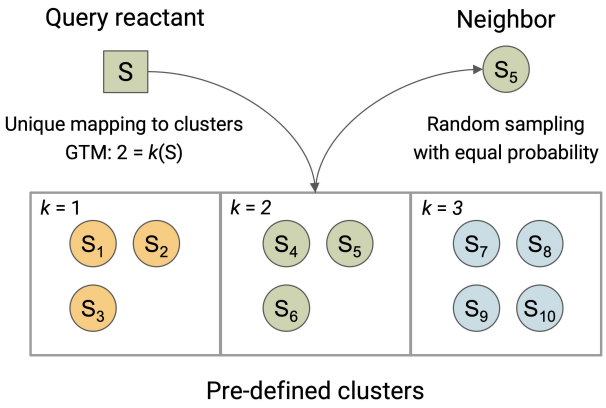


Figure 4.2: Efficient sampling scheme to generate structurally similar reactants based on pre-partitioning of the reactant space. Using a pretrained GTM, any query reactant is uniquely mapped to a cell or cluster to which similar reactants belong. A structurally similar reactant of the query compound can be obtained by sampling the instances in the cell with equal probabilities.

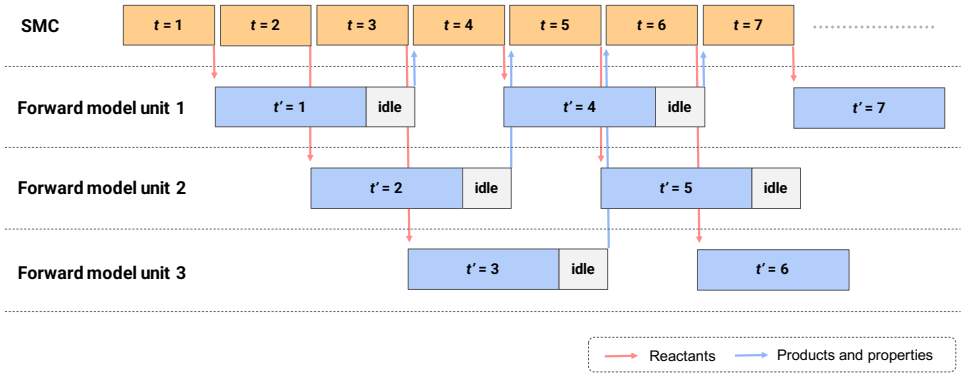


Figure 4.3: SMC algorithm using asynchronous parallel computation. The SMC module proceeds without any suspension (top row). Once the calculation of one step of the SMC is completed, the set of reactants produced there is handed over to the module for the forward calculation in an idle state. The forward-prediction module then calculates the predicted products using Molecular Transformer, which are sent back to the SMC module. Once the data transfer is completed, the forward-prediction module returns to a idle state and waits to receive the next reactant set.

are successively added to the pool of candidate reactants $\mathcal{B}_t$ in the SMC module. The monotonically growing reactant pool $\mathcal{B}_t$ contains the intermediate products calculated from the reaction prediction model. Therefore, throughout the search for single-step reactions in SMC, reaction networks with various structures can be constructed by

sampling and stacking intermediate products for which their reaction networks have already been calculated. The computational complexity of the neighborhood search owing to the monotonic growth of $\mathcal{B}_t$ is suppressed by pre-clustering the set of reactants using the GTM. In summary, the algorithm is based on four ideas: (1) the recurrent algorithm for the network search, which is based on the sequential expansion of the reactant pool, (2) the avoidance of the neighborhood search from large reactant pools using GTM clustering, (3) efficient computation of the forward cascade model using a surrogate model, and (4) the asynchronous parallel computation algorithm.

Algorithm 1 De novo design of functional molecules and their synthetic routes

Initialize commercially available compounds $\mathcal{B} = \{S_1, ...S_N\}$
Initialize intermediate products $\mathcal{I}_0 = \{\}$
Initialize auxiliary distribution $\pi_0(x_0)$,
Initialize number of SMC steps T
Initialize number of particles m
Initialize surrogate forward-prediction model d
Do in multiprocess

SMC

for $t \leftarrow 1$ to T **do**
  $\mathcal{B}_t \leftarrow \mathcal{B} \cup \mathcal{I}_{t'}$
  $X_t = \{\}$
  for $i \leftarrow 1$ to m **do**
    Draw x_t^i from $\eta(x_t|x_{t-1}^i)$
    $X_t \leftarrow X_t \cup \{x_t^i\}$
    Predict surrogate property $Y_t^i = d(\{S_{1,t}^i, S_{2,t}^i\})$
    Calculate the importance weight $w(x_t^i)$ by Y_t^i
    Resample x_t^i from $\pi_t(x_t)$
  end for
  Pass X_t to the forward-prediction module.
end for

Forward prediction

$t' = 0$
for $t' \leftarrow 1$ to T **do**
  Wait for $X_{t'}$
  Assign $X_{t'}$ to a processing unit
  for $x_{t'}^i$ in $X_{t'}$ **do**
    Predict product $P_{t'}^i = r(x_{t'}^i)$
    Predict property $Y_{t'}^i = h(P_{t'}^i)$
  end for
  $\mathcal{I}_{t'} \leftarrow \mathcal{I}_{t'} \cup \{P_{t'}^i | i = 1, \cdots m\}$
  Pass $\mathcal{I}_{t'}$ to the SMC module.
end for

Output: $\{x_t^i | Y_t^i \in U^*; t = 0, \cdots T; i = 1, \cdots m\}$

4.4 Software and potential applications

The analyses shown later can be performed using the Python package Seq-Stack-Reaction (Zhang, 2022), which we have made available online under the license BSD 3-Clause (bsd, 1999). Seq-Stack-Reaction can be installed with Conda (ana, 2020).

Users can plug-in any model for property and synthetic reaction predictions. The list of commercial compounds and the maximum number of reaction steps can also be specified arbitrarily. In the latest version, reactions with a variable number of reactants, such as one- and two-component systems, can be incorporated into the design calculations. In a parallel computing environment, the asynchronous parallel search algorithm can be executed by specifying an option. Additionally, a visualization of the predicted synthetic pathway network of the excavated products is implemented.

In the following, we show an example of applying Seq-Stack-Reaction to the design of drug-like molecules. However, the present method can be applied to general molecular design tasks in materials research. To demonstrate the versatility of the proposed method, as an additional example, the GitHub website provides a sample dataset and Python scripts to design lubricant molecules by following the task designed in (Kajita et al., 2020). As the target property, we choose the viscosity index (VI) that indicates the temperature sensitivity of viscosity. To maintain stable machine operations, oil with a high VI value is required by machinery equipment. Because the VI is known to underestimate the viscosity susceptivity of low-viscosity oils, a complementary index called the dynamic viscosity index (DVI) was also added to the design goal. Using a structure-property dataset obtained from all-atom classical molecular dynamics simulations in (Kajita et al., 2020), we predicted the VI and DVI from the chemical structure of any given lubricant molecule. Its inverse map was then computed to identify highly viscous molecules with their synthetic pathways that could be designed from a given set of reactants.

Chapter 5

Experiments

The predictive and computational performances of the proposed method were evaluated using an application example that involved designing drug-like molecules. In particular, we constructed several variants of the Bayesian molecular design algorithms by combining the four constituent mechanisms described in the previous section, and compared their performance to quantitatively investigate their individual contributions to the overall scheme.

5.1 Target properties

As target properties $Y = h(P)$, we considered the following two physicochemical properties that quantitatively express the drug-likeliness of a designed molecule P.

- **QED** The quantitative estimate of drug-likeness (QED) quantifies drug-likeness as a score ranging between 0 and 1 (Bickerton et al., 2012). QED was modeled on a dataset of 771 known oral drugs using eight descriptors, including molecular weight, polar surface area, and number of hydrogen bond donors and acceptors. The higher the QED value, the more drug-like the molecule is judged to be. The target range of the QED was set to be greater than 0.8.

- **logP** The octanol-water partition coefficient logP is defined as the normal logarithm of the ratio of the concentrations of molecules in the organic and aqueous layers at equilibrium. According to Lipinski's rule of five (Lipinski et al., 1997), the target range of logP was defined as not exceeding 5.

Computational models of these two properties are implemented in the modules of RDKit (Landrum et al., 2006).

5.2 Reaction prediction

As a forward response prediction model $P = g(S)$, we employed Molecular Transformer (Schwaller et al., 2019), which is known to be a standard model. This attention-based neural translation model defines a translation between the SMILES strings of reactants and their products. For simplicity, reagents were removed from the model input. SMILES strings for multiple reactants were input into the model as separated by periods ".". The SMILES strings were all canonicalized using RDKit. The inputs were tokenized with the regular expression according to the original paper of Molecular Transformer.

The model was trained from scratch using 80% of the training instances randomly selected from the 500k recorded reactants and products in the USPTO dataset (Lowe, 2012). The top 1 prediction accuracy of the trained model for the remaining data reached 78.2%, which is comparable to the accuracy reported in previous studies (Guo et al., 2020; Pesciullesi et al., 2020).

5.3 Surrogate models

We applied gradient boosting regression to construct surrogate models that predict the two target properties from a set of two reactants without going through the prediction of synthetic products using Molecular Transformer. We randomly selected two pairs of 492K reactants from USPTO, generated their products using Molecular Transformer, and calculated their QED and logP. The resulting set of 394K samples was used to train the surrogate models for QED and logP. The R^2 values for predicting QED and logP for the 98K additional test samples were 0.78 and 0.86, respectively.

5.4 Commercial compounds

We used a subset of the Enamine building block catalog global stock as the set of commercially available reactants by which virtual molecules are synthesized (ena, 2020). This set, consisting of 150K unique building blocks, was narrowed down by Gottipati et al. (2020) to those applicable to one or more rules in a template-based reaction prediction model. The design task is to identify the synthesizable products in this reactant set and to meet the required properties. The Enamine reactant set was also used to train the GTM model as well. Each reactant was transformed into a binary vector of length 3239 concatenated with the Morgan fingerprint of radius 2 and bit length 1024, the MACCS key, and the RDKit fingerprint. A total of 441 predefined clusters were used in the similarity-based SMC proposal.

5.5 Sequential Monte Carlo

As explained previously, the proposed method was designed based on four ideas. To investigate the contribution of each idea to the overall performance, we implemented the five variants listed in Table 5.1: (1) SMC-RECUR-GTM-SR-PL, (2) SMC-RECUR-GTM, (3) SMC, (4) Random, and (5) Random-RECUR. The meanings of the abbreviations attached to each method are as follows:

- **RECUR** Recurrent design technique for the reaction networks
- **GTM** The use of the 441 GTM clusters in the similarity-based proposal
- **SR** Surrogate models for the direct prediction of QED and logP
- **PL** Asynchronous parallel computing of the SMC and reaction prediction modules

For the vanilla SMC in (3), only single-chain synthetic reactions with the number of steps fixed at $n \in \{1, 2, 3\}$ were considered. For example, in the case of $n = 2$, we considered only the cascade-type reactions as $S_{11} + S_{12} \rightarrow P_1, P_1 + S_{21} \rightarrow P_2$ and searched for three reactants $S = \{S_{11}, S_{12}, S_{21}\}$ that synthesize product $P = P_2$ satisfying the property requirements. For "Random" in (4), a completely random search for two reactants was conducted for the single-step reaction $S_{11} + S_{12} \rightarrow P_1$. The variant "Random-RECUR" in (5) represents an integrated method of random search and recursive network design algorithm.

The experimental conditions were set to be common to the five variants: for SMC and Random search, the number of iterations was set to $T = 10000$, the number of particles was set to $m = 500$, for SMC, the exploration–exploitation trade-off parameter of the proposal distribution was set to $\alpha = 0.8$, indicating a 20% chance of making an exploratory search.

5.6 Computational environments

The experiments were carried out on an NVIDIA DGX STATION with 4 Tesla V100 and Intel Xeon E5-2698 v4 CPUs.

5.7 Results

After the experiment, we compared the five methods listed in Table 5.1. As mentioned above, for the vanilla SMC, the design of the single-chain reaction routes was tested under three different step sizes $n \in \{1, 2, 3\}$. Therefore, seven methods were included

Table 5.1: Five different algorithms to be evaluated for the performance test. The abbreviations are explained in the main text.

Method	SMC	Network type	Parallel
SMC-RECUR-GTM	✓	any	-
SMC-RECUR-GTM-SR-PL	✓	any	✓
SMC-n	✓	n-step single chain ($n \in \{1, 2, 3\}$)	-
Random	-	one-step single chain	-
Random-RECUR	-	any	-

in this comparison. Each method was tested in two scenarios. In scenario 1, a relatively small number of commercial compounds was assumed to be available: the candidate reactants were generated by randomly sampling 10k of the 150k Enamine compounds available for purchase. In scenario 2, all 150k compounds were used to design the molecules.

First, we report the results of scenario 1. As a performance measure, we simply considered the number of unique designed molecules that reached the target property range. Figure 5.1 shows the evolution of the number of unique hit molecules at each step of the sequential calculation. Unsurprisingly, the two random search algorithms were clearly less efficient than the others. For SMC-1 and Random, whose design space is restricted to single-step reactions with two reactants, the number of hits decayed as the number of search steps increased. This means that the limited design space consisting of two combinations of at-most 10k commercial compounds can be adequately covered by the naive algorithms. In contrast, for the vanilla SMC with two- and three-step reactions (SMC-2, SMC-3) and the recurrent algorithms to explore arbitrary reaction networks (SMC-RECUR-GTM, SMC-RECUR-GTM-SR-PL), the number of hit molecules at each step remained constant, and no decay trend was observed for $T = 10000$. This observation confirms that the search space for general reaction design is quite large. In the comparison between SMC-RECUR-GTM and its extended version, SMC-RECUR-GTM-SR-PL with the surrogate models shows that the number of hits of the latter is slightly lower than that of the former. This is a natural consequence because the surrogate models involve approximation errors.

Figure 5.2 shows the evolution of the cumulative number of unique hit molecules that reached the target region against the number of steps. The left panel shows the cumulative number of hits as a function of the total number of molecules generated. This is a different view of the results in Figure 5.1. It was confirmed that SMC-RECUR-GTM is the most efficient method to find molecules hidden in the target region. It was also confirmed that SMC-RECUR-GTM-SR-PL has a reduced hit rate due to

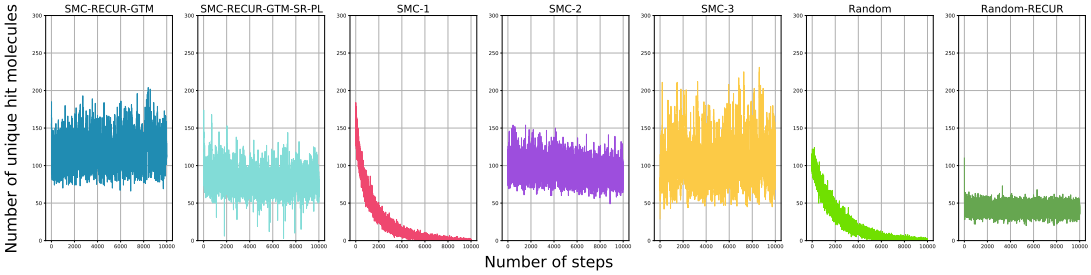


Figure 5.1: Number of design molecules reaching the target property range at each step for scenario 1 where the number of commercially available reactants is limited to 10K. The correspondence between the seven methods and their labels is shown in Table 5.1.

the use of the surrogate models involving approximation errors, as mentioned above. Vanilla SMC-2 and SMC-3 also consistently found the target molecules. This suggests that it is difficult to fully cover the large design space with as few as $T \times m = 500000$ search trials. The right panel in Figure 5.2 shows the cumulative number of hits per step as a function of execution time (right panel). In terms of computational cost, the parallel recursive molecular design algorithm SMC-RECUR-GTM-SR-PL with the surrogate models showed by far the highest search efficiency.

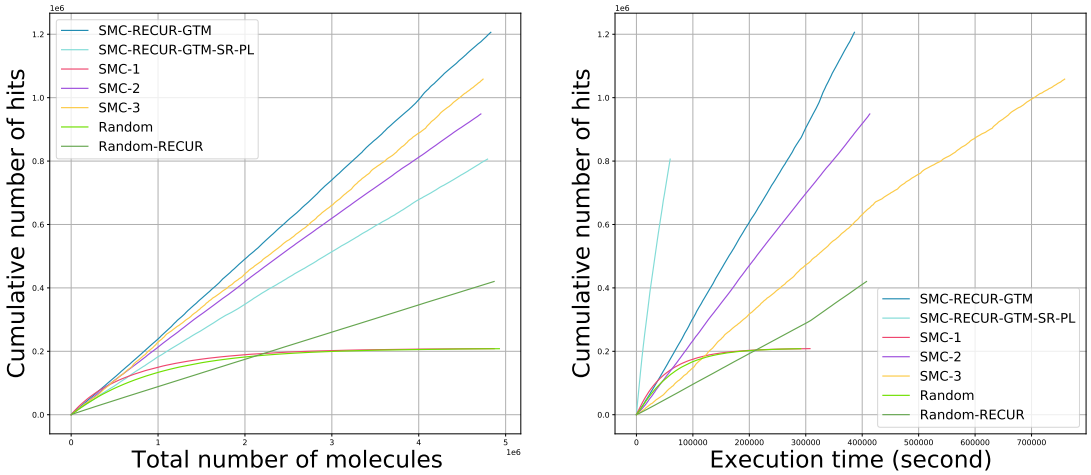


Figure 5.2: Cumulative number of hit molecules in scenario 1 where the number of commercially available reactants is limited to 10K. The left and right plots show the cumulative number of hits as a function of the total number of molecules generated and the execution time (CPU time), respectively.

With regard to the results for scenario 2: The number of hit molecules at each step

(Figure 5.3) and the cumulative number of hits (Figure 5.4) were significantly lower for the two random searches, as in scenario 1. In contrast to the results of scenario 1, SMC-1 and Random, whose design space was restricted to single-step reactions with two reactants, did not show any decay in the number of hit molecules throughout all steps. This observation indicates that, as the number of commercial reactants increases, the design space becomes much larger, even for single-step reactions. Among the methods other than the random searches, no significant difference was found in the evolution of the number of hit molecules and the cumulative number. However, similar to the results of scenario 1, SMC-RECUR-GTM-SR-PL showed by far the highest search efficiency in terms of the number of hit molecules per execution time (right panel in Figure 5.4). From a practical point of view, search performance against execution time is considered the most important criterion. Therefore, we conclude that SMC-RECUR-GTM-SR-PL is superior in terms of the evaluation criteria for detecting the number of unique molecules.

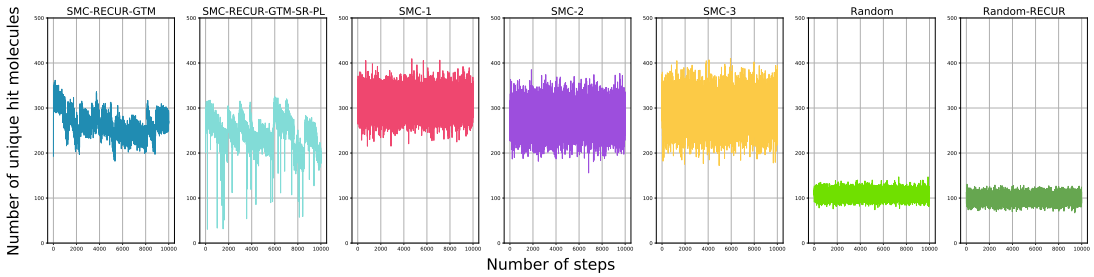


Figure 5.3: Number of design molecules reaching the target property range at each step for scenario 2 where all 150K commercial compounds were used in the design.

We then performed a quality assessment based on the structural diversity and novelty of the generated hit molecules and their coverage against existing molecules synthesized so far. Specifically, we assessed the coverage and novelty of the 300K hypothetical molecules that reached the target region in scenario 2 against a hit compound set extracted from the organic compound database ChEMBL (Gaulton et al., 2017) as follows:

(1) The set of hit compounds, denoted by A, was obtained by calculating the two properties of 127k compounds registered in the ChEMBL.

(2) Let B be the set of 300K hit virtual molecules generated in scenario 2.

(3) Evaluate the similarity between the compounds in A and B using the Tanimoto coefficient of the ECFP fingerprint descriptor (radius 2, bit length 2048).

(4) Set the threshold values of the Tanimoto coefficient as $\gamma \in \{0, 0.1, ..., 1\}$.

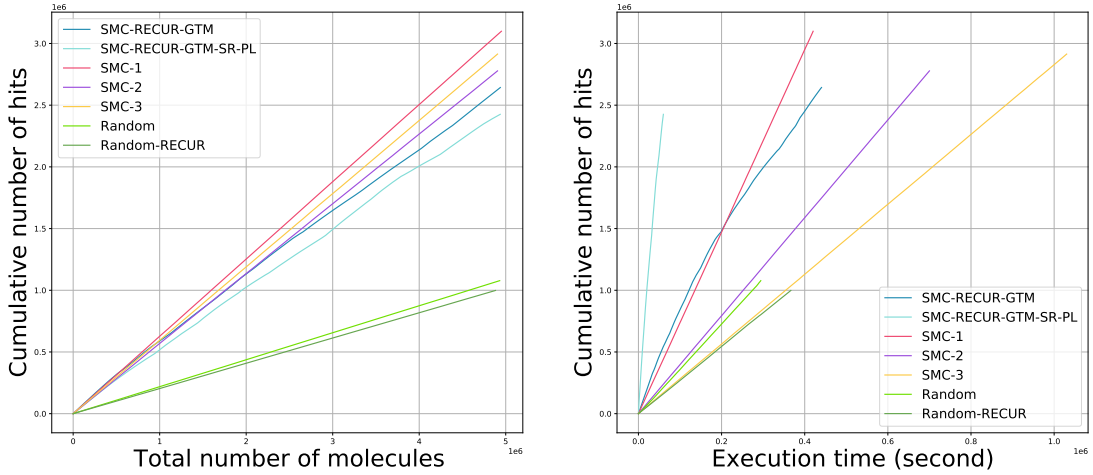


Figure 5.4: Cumulative number of hit molecules in scenario 2 where all 150K commercial compounds were used in the design. The left and right plots show the cumulative number of hits as a function of the total number of molecules generated and the execution time (CPU time), respectively.

(5) Coverage: Calculate the percentage of molecules in A with a similarity greater than γ to those in B.

(6) Novelty: Calculate the percentage of molecules in B with a similarity smaller than γ to those in A.

(7) Vary the threshold γ from 0 to 1, and draw a curve representing the balance between coverage and novelty (CN curve), as in Figure 5.5.

The CN curve shows an upward or downward convex pattern depending on the inclusive relationship of the distributions of A and B. Ideally, a set of reasonably novel and diverse molecules should be generated, while maintaining a high coverage to existing molecules. This corresponds to a situation in which the distribution of B encompasses A. In such a case, the CN curve deviates slightly from the 45° line and shows an upward convex pattern. For this criterion, SMC-RECUR-GTM, SMC-RECUR-GTM-SR-PL, and Random-RECUR showed better properties than the others (Figure 5.5). When the threshold of Tanimoto similarity was set to $\gamma \geq 0.7$, the coverage and novelty of SMC-RECUR-GTM and SMC-RECUR-GTM-SR-PL were both approximately 0.3 and 0.8, respectively.

Note that the SMC-3 showed a significantly lower novelty than the others. As the search space grows, the ordinary SMC is more likely to get trapped in local solutions, making it more difficult to escape from them in a finite run time. The increase in the

number of steps in the single-chain reaction from two to three caused the SMC to significantly degrade in performance.

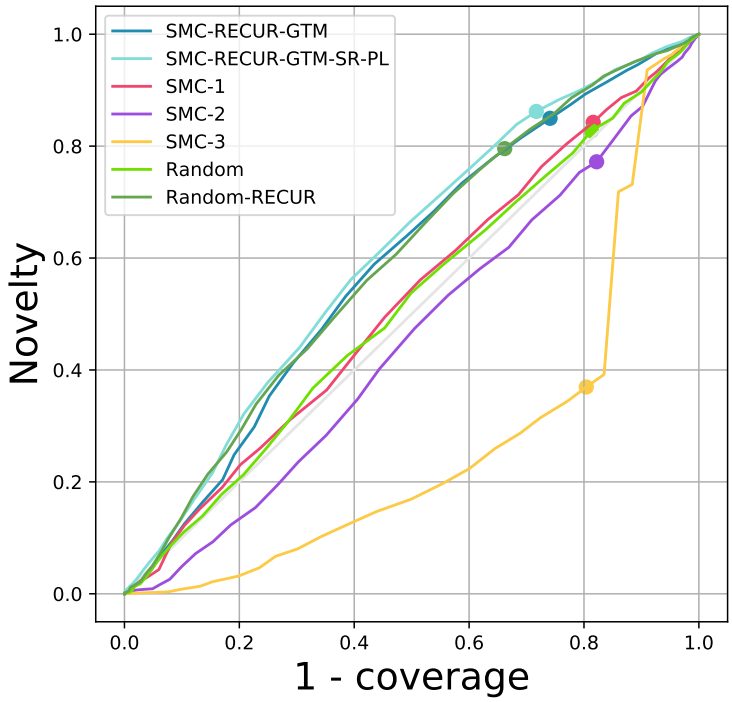


Figure 5.5: 1-coverage (horizontal axis) and novelty (vertical axis) of the set of designed virtual molecules with respect to existing molecules in ChMBLE, which is drawn as a function of varying thresholds of Tanimoto similarity. The circle represents the coverage and novelty when the similarity threshold is set to 0.7.

Table 5.2 provides a summary of the performance comparison between the proposed method (SMC-RECUR-GTM and SMC-RECUR-GTM-SR-PL) and eight commonly used molecular generators: SMILES-LSTM (Segler et al., 2018a), Character VAE (CVAE) (Gómez-Bombarelli et al., 2018), Grammar VAE (GVAE) (Kusner et al., 2017), GraphVAE (Simonovsky & Komodakis, 2018), Junction Tree Autoencoder (JT-VAE) (Jin et al., 2018), Constrained Graph Autoencoder (CGVAE) (Liu et al., 2018), Molecule Chef (Bradshaw et al., 2019), and DoG-Gen (Bradshaw et al., 2020). Here, we have reported six performance metrics (validity, uniqueness, quality, novelty, FCD, and synthetic accessibility score) as a common benchmark proposed by Brown et al. (2019) that were reported in the previous studies:

- **Validity** Proportion of generated molecules that can be parsed by RDKit
- **Uniqueness** Proportion of unique molecules that can be parsed by RDKit

- **Novelty** Proportions of molecules that are different from the training instances
- **Quality** Compound quality measurements, a rule set is employed to decide whether compounds can be included in high throughput screening
- **FCD** Fréchet ChemNet distance between generated molecules and a predefined set of existing molecules
- **SA score** The synthetic accessibility (SA) score (Ertl & Schuffenhauer, 2009)

Note that these reported values were calculated on different tasks in the different studies. Although the result cannot be used to discuss superiority or inferiority based on performance values, it is confirmed that the performance values achieved by the proposed method are at almost the same level as those of the related methods.

To evaluate the synthesizability of the designed molecules, we used the SA score, which takes values from 1 to 10, with higher values indicating more difficult to synthesize. To distinguish between compounds that are easy and difficult to synthesize, a threshold value of 6.0 has empirically been used. Using RDKit, we calculated the SA score for 20K molecules randomly selected from the hit compounds of SMC-RECUR-GTM-SR-PL. Their histogram is shown and compared with one of the existing molecules in ChEMBL in Figure 5.6. Most SA scores are concentrated at 3.2, indicating that their distribution is not significantly different from that of previously synthesized molecules.

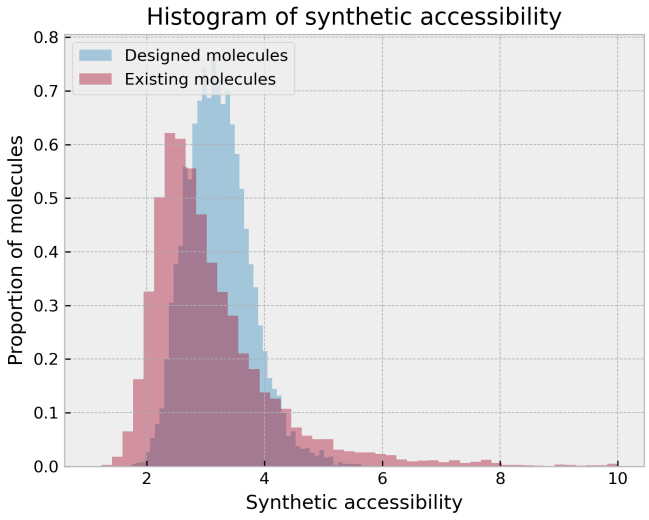


Figure 5.6: Histograms of SA scores of hit compounds obtained from the proposed method and existing molecules in ChEMBL.

Table 5.2: Validity, uniqueness, quality, FCD, and the mean of the average SA score of the hit compounds generated from the proposed method and the performance values of other methods reported in previous studies. Note that the reported values were obtained from different tasks and experimental conditions.

Method	Task	Validity	Uniqueness	Novelty	Quality	FCD	SA
SMC-RECUR-GTM	QED, logP	100.0	97.6	98.9	86.78	0.78	3.27
SMC-RECUR-GTM-SR-PL	QED, logP	100.0	98.7	98.7	86.16	0.77	3.27
DoG-Gen (Bradshaw et al., 2020)	Structural features, physicochemical properties, etc.	100.0	97.7	88.4	101.6	0.45	3.31
SMILES LSTM (Segler et al., 2018a)	serotonin receptor, plasmodium falciparum, and Staphylococcus aureus	94.8	95.5	74.9	101.93	0.46	3.14
CVAE (Gómez-Bombarelli et al., 2018)	QED, SA score	96.2	97.6	76.9	103.82	0.43	3.22
GVAE (Kusner et al., 2017)	logP	74.4	97.8	82.7	98.98	0.89	2.98
GraphVAE (Simonovsky & Komodakis, 2018)	QM9 (Ramakrishnan et al., 2014) and ZINC (Irwin et al., 2012)	42.2	57.7	96.1	94.64	13.92	3.29
JT-VAE (Jin et al., 2018)	logP	100.0	99.2	94.9	102.34	0.93	3.32
CGVAE (Liu et al., 2018)	QED	100.0	97.8	97.9	45.64	14.26	3.19
Molecule Chef (Bradshaw et al., 2019)	QED	98.9	96.7	90.0	99.0	0.79	3.25

Here, we remark on the chemical validity of the predicted reaction networks. To see this, three examples of predicted reaction networks are shown in Figures 5.7–5.9, which were drawn with the Seq-Stack-Reaction library. Each reaction in the three reaction pathways was assessed by our expert chemist, in which unknown reagents and reaction conditions were inferred based on expert knowledge. The validity was classified into one of $\{1, 2, 3\}$ as follows: 1, feasible with suitable catalysts and/or reagents; 2, difficult to react due to low reactivity or other factors; 3, infeasible. Detailed explanations have been provided in the figure captions. The rationale for each scoring is given in the figure captions. Overall, half of the predicted reactions are considered possible; the remaining reactions are considered less reactive or infeasible.

Further, the SA score was used to evaluate the synthesizability of an intermediate and final product in each reaction step. According to this evaluation criterion, most of the products on the designed network were judged to be synthesizable. However, it should be noted that the SA score is a measure of the synthesizability of a product molecule and does not evaluate the feasibility of each reaction. In fact, although the three reaction networks contained apparently infeasible reaction steps, the synthesizability of their intermediate products was determined to be high according to the SA scores. The accuracy of the forward reaction predictor will have to be greatly improved in order to stably present a chemically valid synthetic reaction network. Therefore, while the current methodology is useful as a decision-support tool to encourage expert ideas, it cannot be used for predictive models such as those implemented in fully automated chemical synthesis.

For further validation, it is necessary to develop methods and platforms to systematically evaluate the properties of designed molecules and the feasibility of designed synthetic pathways. The computation of molecular property characterization and synthetic reactions using quantum chemical methods is extremely time-consuming; therefore, the implementation of a high-throughput evaluation system is quite challenging. The establishment of a systematic validation method is an important issue for future research.

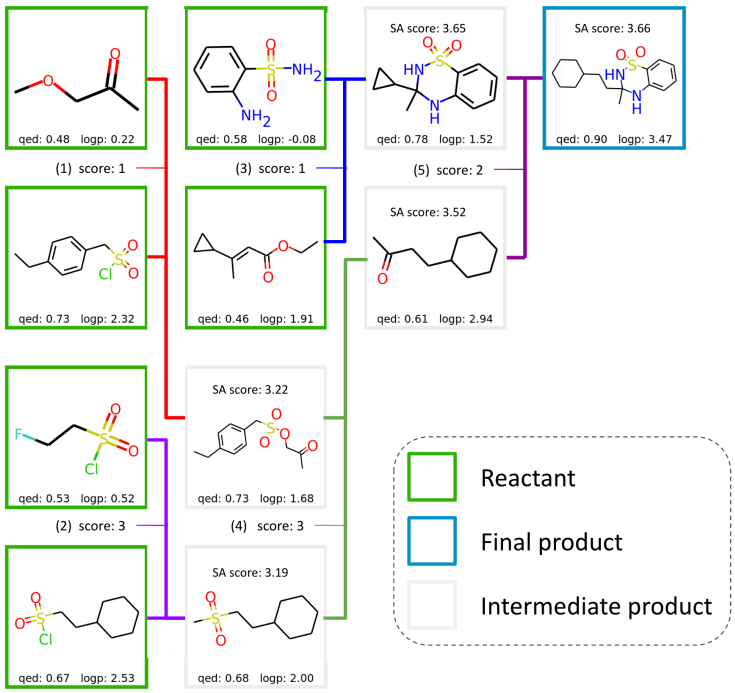


Figure 5.7: A synthetic reaction network consists of 5 single-step reactions (1)–(5). The SA scores are given to the intermediate and final products. The validity of each reaction was classified into one of $\{1, 2, 3\}$ as follows: 1, feasible; 2, difficult to react due to low reactivity or other factors; 3, infeasible. (1) This reaction is feasible with a suitable reagent that hydrolyzes the upper reactant to alcohol, the reaction between the alcohol and sulfonyl chloride (R–SO_2–Cl) is often used in the synthesis of sulfonate esters (R–SO_2–OR'). (2) This is infeasible because the methyl group cannot be found in the reactants. (3) The lower reactant is an α, β-unsaturated carbonyl, which is susceptible to attack by nucleophiles such as an amino group at the β-carbon. (4) This would be difficult, considering the general reaction of sulfonate esters, $CH_2C(=O)CH_3$ is released from the sulfonate ester. The $CH_2C(=O)CH_3$ and CH_3SO_2 in the lower reactant may undergo a substitution reaction with some catalysts, but in this case, the number of carbons in the product is incorrect. (5) The reaction sites are saturated C–C bonds, which generally exhibits low reactivity.

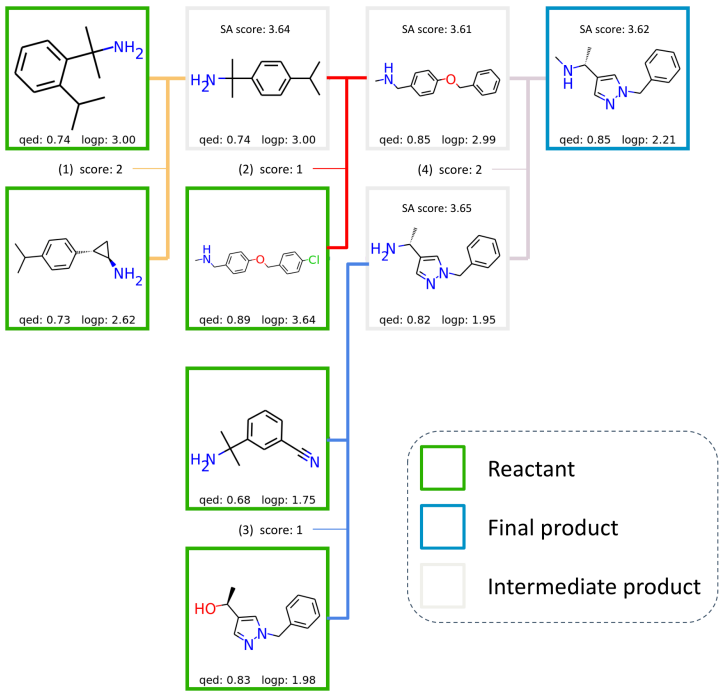


Figure 5.8: A synthetic reaction network consists of 4 single-step reactions (1)–(4). (1) The reaction sites are saturated C–C bonds, which generally exhibits low reactivity. (2) This reaction is feasible by using Fe catalysts that can dechlorinate chlorobenzene. In this reaction, the upper reactant is not necessary. (3) This reaction is feasible by using reagents that can release NH_2 anion from the upper reactant. The NH_2 anion can substitute for the OH group in the lower reactant by S_N2 reactions because the NH_2 anion has higher nucleophilicity than OH anion, and chiral inversion occurs in the product. (4) The reaction is difficult to proceed. This product can be synthesized from iodomethane and the lower reactant.

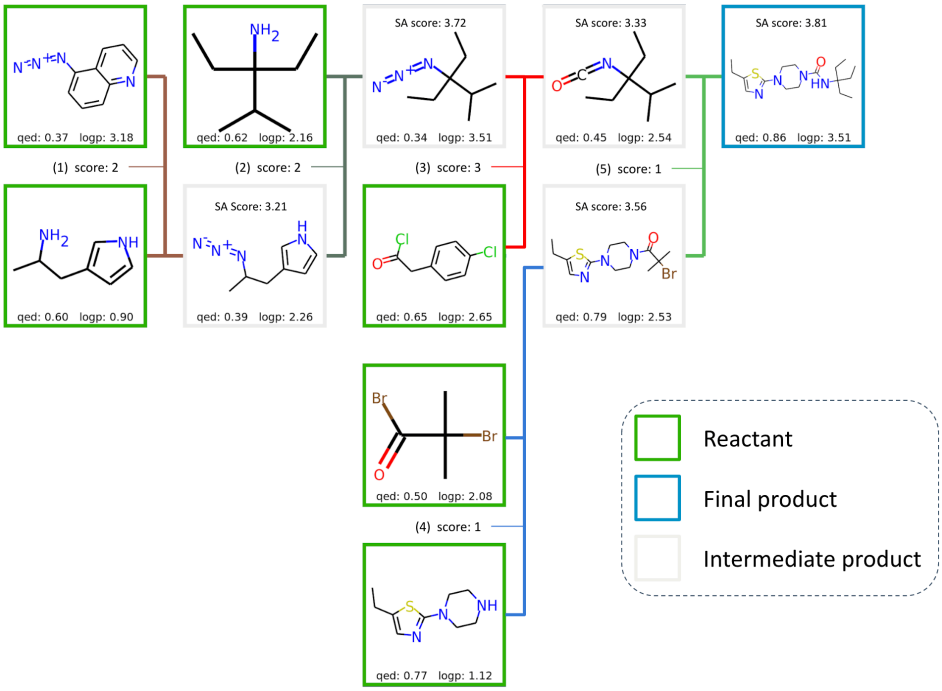


Figure 5.9: A synthetic reaction network consists of 5 single-step reactions (1)–(5). (1)–(2) Diazo transfer reactions. Reactants with an azide group (-N_3) generally exhibit low reactivity for diazo transfer reactions. CF_3SO_2-N_3 is commonly used as an azide reagent in diazo transfer reactions. (3) The azide reagents (upper reactant) and acyl chlorides (R-C(=O)Cl in the lower reactant) can undergo the Curtius rearrangement, leading to isocyanates (R-NCO). However, this reaction is incorrect because the Curtius rearrangement should convert the acyl chloride group in the lower reactant to the isocyanate group. (4) The acyl bromide group (C(=O)–Br) in the upper reactant can undergo nucleophilic attack from the secondary amine group in the lower reactant, leading to the amide group (R–NC(=O)–R') in the product. (5) This is feasible with a suitable reagent that hydrolyzes the lower reactant to the secondary amine. The amines and isocyanates react to produce the urea product (R–N–C(=O)–NH–R'). Note that in this case, reaction (4) is not necessary because the final product can be synthesized from the isocyanate reactant (the upper reactant in reaction (5)) and the lower reactant in reaction (4).

Chapter 6

Conclusions and future works

6.1 Limitation

In this study, we put forth a set of novel ideas aimed at addressing a crucial issue in the domain of molecular design. However, it is important to acknowledge that our proposed method is not devoid of limitations, which warrant careful consideration. One of the limitations is the Molecular Transformer, Molecular Transformer is contingent upon the availability and quality of the training data. This model excels in situations where the input data closely resembles the molecular structures it has been trained on. This limitation is inherent in its training paradigm, as it learns patterns and relationships from a specific dataset. Consequently, the model's performance might falter when confronted with molecules that exhibit structural variations or features outside the scope of its training data.

Another significant limitation pertains to the cumulative accuracy of reaction prediction. Given that the predictive performance of the one-step reaction prediction model is not flawless, the repeated utilization of this model along an extended reaction pathway introduces the potential for error accumulation. Consequently, this cumulative error poses a significant challenge, ultimately resulting in the generation of infeasible final product outcomes.

Moreover, there is a challenge in the validation step. In our pursuit of refining the accuracy and reliability of our molecular design process, we engaged chemists to provide expert judgments on each individual step. Their extensive domain knowledge and expertise play a crucial role in evaluating the feasibility, efficiency, and practicality of the proposed molecular designs. This collaborative approach represents an effort to bridge the gap between machine-generated suggestions and human intuition, leveraging the strengths of both sides. However, it's important to acknowledge that while chemists' insights are invaluable, conducting systematic validation can be challenging. The complexity of molecular interactions, the vastness of chemical space, and the

intricacies of reaction mechanisms all contribute to the intricacy of the validation process. While chemists' assessments offer qualitative feedback, a more quantitative and comprehensive validation framework is required to ensure the robustness and accuracy of our molecular designs.

6.2 Future works

Up until this point, our focus in this endeavor has primarily centered on specific aspects of molecular design, notably excluding catalysts and reaction conditions. Acknowledging the pivotal role that catalysts and reaction conditions play in influencing the outcome of chemical reactions, we recognize the potential benefits of integrating these factors into our molecular design framework. Recent advancements in the field have demonstrated that harnessing this additional information can yield more refined and contextually relevant molecular designs. In light of this, our future trajectory involves incorporating catalysts and reaction conditions into our design process to extract enhanced insights and produce more effective solutions.

Moreover, extending the scope of our proposed method, we envision its application in the generation of polymer libraries. As part of our research initiatives, our team has been diligently developing a rule-based polymer generator (Ohno et al., 2023 in press). This work aims to enhance the diversity and versatility of polymers that can be synthesized for various applications. The proposed method in this paper is expected to make full use of the new polymer generator.

6.3 Summary

We present a machine-learning methodology and a general-purpose Python library for the simultaneous design of molecules with desired properties and their synthetic reactions with any network topology. Our methodology involves constructing a forward model using synthetic reaction prediction models and property prediction models as building blocks. The forward model predicts the properties of the final products that result from a given reaction network and reactant set. We then obtain the inverse mapping based on a Bayesian inference framework to simultaneously identify a reaction network, its constituent reactant sets, and the final products that satisfy arbitrary target properties. The reactant set is selected based on a combination of predefined commercial compounds. The design space for our methodology consists of arbitrary reaction networks and reactant sets, which are very large. Therefore, we developed a sequential Monte Carlo algorithm incorporating a recurrent algorithm for the network search. This algorithm was designed to efficiently explore the design space and identify high-quality virtual molecules that meet the desired properties. We

conducted performance tests to assess the ability of our algorithm to find high-quality virtual molecules. The quality of the designed molecules was evaluated based on their reproducibility and novelty with respect to previously synthesized molecules. Our algorithm successfully generated high-quality virtual molecules that are both reproducible and novel with respect to previously synthesized molecules. We developed a distributed Python library for our machine learning methodology, which provides an interface that allows users to plug in arbitrary reaction prediction models, property prediction models, and a set of commercial compounds. The library is designed to be modular and extensible, allowing users to easily add their own models and compounds. It also includes a set of pre-trained models for reaction and property prediction, which can be used out-of-the-box for many applications. The library is expected to facilitate the widespread use of our methodology in practical applications. The use of machine learning in molecular design has shown great promise in recent years. Our methodology provides a flexible and efficient approach to the simultaneous design of molecules and their synthetic reactions. Our approach is particularly advantageous for large and complex design spaces, where traditional methods may be impractical. Our methodology also provides a useful tool for discovering novel molecules with desired properties, which can have important applications in drug design, materials science, and many other fields.

Although machine learning-based research for molecular design and synthetic pathway design has been actively pursued in recent years, most such studies have worked independently on the two subjects so far. Thus, research on the simultaneous design of functional molecules and synthetic pathways has not progressed significantly. In particular, few general-purpose libraries are currently available. The aim of this study was to develop a generic methodology and tools to link these two subjects. We believe that this milestone has been achieved.

References

BSD 3-Clause, 1999. URL https://opensource.org/licenses/BSD-3-Clause.

Anaconda Software Distribution, 2020. URL https://docs.anaconda.com/.

Enamine Building Block catalogue, 2020. URL https://enamine.net/mail/bb-2003.html.

Jeremy Ash and Denis Fourches. Characterizing the chemical space of ERK2 kinase inhibitors using descriptors computed from molecular dynamics trajectories. *Journal of Chemical Information and Modeling*, 57(6):1286–1299, 2017.

Rim Assouel, Mohamed Ahmed, Marwin H Segler, Amir Saffari, and Yoshua Bengio. Defactor: Differentiable edge factorization-based probabilistic graph generation. *arXiv preprint arXiv:1811.09766*, 2018.

Pierre Baldi. Autoencoders, unsupervised learning, and deep architectures. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, pp. 37–49. JMLR Workshop and Conference Proceedings, 2012.

G Richard Bickerton, Gaia V Paolini, Jérémy Besnard, Sorel Muresan, and Andrew L Hopkins. Quantifying the chemical beauty of drugs. *Nature Chemistry*, 4(2):90–98, 2012.

Christopher M Bishop, Markus Svensén, and Christopher KI Williams. GTM: The generative topographic mapping. *Neural Computation*, 10(1):215–234, 1998.

John Bradshaw, Brooks Paige, Matt J Kusner, Marwin Segler, and José Miguel Hernández-Lobato. A model to search for synthesizable molecules. In *Advances in Neural Information Processing Systems*, pp. 7937–7949, 2019.

John Bradshaw, Brooks Paige, Matt J Kusner, Marwin Segler, and José Miguel Hernández-Lobato. Barking up the right tree: an approach to search over molecule synthesis dags. *Advances in Neural Information Processing Systems*, 33:6852–6866, 2020.

Nathan Brown, Marco Fiscato, Marwin HS Segler, and Alain C Vaucher. GuacaMol: benchmarking models for de novo molecular design. *Journal of Chemical Information and Modeling*, 59(3):1096–1108, 2019.

Tim Capes, Paul Coles, Alistair Conkie, Ladan Golipour, Abie Hadjitarkhani, Qiong Hu, Nancy Huddleston, Melvyn Hunt, Jiangchuan Li, Matthias Neeracher, et al. Siri on-device deep learning-guided unit selection text-to-speech system. In *Interspeech*, pp. 4011–4015, 2017.

Hanjun Dai, Yingtao Tian, Bo Dai, Steven Skiena, and Le Song. Syntax-directed variational autoencoder for structured data. *arXiv preprint arXiv:1802.08786*, 2018.

Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential monte carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3): 411–436, 2006.

Joseph L Durant, Burton A Leland, Douglas R Henry, and James G Nourse. Reoptimization of MDL keys for use in drug discovery. *Journal of Chemical Information and Computer Sciences*, 42(6):1273–1280, 2002.

Peter Ertl and Ansgar Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of Cheminformatics*, 1:1–11, 2009.

Wenhao Gao, Rocío Mercado, and Connor W Coley. Amortized tree generation for bottom-up synthesis planning and synthesizable molecular design. *arXiv preprint arXiv:2110.06389*, 2021.

Anna Gaulton, Anne Hersey, Michał Nowotka, A Patricia Bento, Jon Chambers, David Mendez, Prudence Mutowo, Francis Atkinson, Louisa J Bellis, Elena Cibrián-Uhalte, et al. The ChEMBL database. *Nucleic Acids Research*, 45(D1):D945–D954, 2017.

Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, 2018.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Sai Krishna Gottipati, Boris Sattarov, Sufeng Niu, Yashaswi Pathak, Haoran Wei, Shengchao Liu, Simon Blackburn, Karam Thomas, Connor Coley, Jian Tang, et al. Learning to navigate the synthetically accessible chemical space using reinforcement learning. In *International Conference on Machine Learning*, pp. 3668–3679. Proceedings of Machine Learning Research, 2020.

Zhongliang Guo, Stephen Wu, Mitsuru Ohno, and Ryo Yoshida. Bayesian algorithm for retrosynthesis. *Journal of Chemical Information and Modeling*, 60(10):4474–4486, 2020.

Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6): 82–97, 2012.

Huixiao Hong, Qian Xie, Weigong Ge, Feng Qian, Hong Fang, Leming Shi, Zhenqiang Su, Roger Perkins, and Weida Tong. Mold2, molecular descriptors from 2D structures for chemoinformatics and toxicoinformatics. *Journal of Chemical Information and Modeling*, 48(7):1337–1344, 2008.

Julien Horwood and Emmanuel Noutahi. Molecular design in synthetically accessible chemical space via deep reinforcement learning. *ACS omega*, 5(51):32984–32994, 2020.

Hisaki Ikebata, Kenta Hongo, Tetsu Isomura, Ryo Maezono, and Ryo Yoshida. Bayesian molecular design with a chemical language model. *Journal of Computer-aided Molecular Design*, 31(4):379–391, 2017.

John J Irwin, Teague Sterling, Michael M Mysinger, Erin S Bolstad, and Ryan G Coleman. ZINC: a free tool to discover chemistry for biology. *Journal of Chemical Information and Modeling*, 52(7):1757–1768, 2012.

Wengong Jin, Connor Coley, Regina Barzilay, and Tommi Jaakkola. Predicting organic reaction outcomes with Weisfeiler-Lehman network. In *Advances in Neural Information Processing Systems*, volume 30, pp. 2607–2616, 2017.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *International Conference on Machine Learning*, pp. 2323–2332. Proceedings of Machine Learning Research, 2018.

Artur Kadurin, Alexander Aliper, Andrey Kazennov, Polina Mamoshina, Quentin Vanhaelen, Kuzma Khrabrov, and Alex Zhavoronkov. The cornucopia of meaningful

leads: Applying deep adversarial autoencoders for new molecule development in oncology. *Oncotarget*, 8(7):10883, 2017.

Hiroshi Kajino. Molecular hypergraph grammar with its application to molecular optimization. In *International Conference on Machine Learning*, pp. 3183–3191. Proceedings of Machine Learning Research, 2019.

Seiji Kajita, Tomoyuki Kinjo, and Tomoki Nishi. Autonomous molecular design by Monte-Carlo tree search and rapid evaluations using molecular dynamics simulations. *Communications Physics*, 3(1):77, 2020.

Matt J Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder. In *International Conference on Machine Learning*, pp. 1945–1954. PMLR, 2017.

Greg Landrum et al. RDKit: Open-source cheminformatics. 2006.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Yujia Li, Oriol Vinyals, Chris Dyer, Razvan Pascanu, and Peter Battaglia. Learning deep generative models of graphs. *arXiv preprint arXiv:1803.03324*, 2018.

Kangjie Lin, Youjun Xu, Jianfeng Pei, and Luhua Lai. Automatic retrosynthetic route planning using template-free models. *Chemical Science*, 11(12):3355–3364, 2020.

Christopher A Lipinski, Franco Lombardo, Beryl W Dominy, and Paul J Feeney. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Advanced Drug Delivery Reviews*, 23 (1-3):3–25, 1997.

Bowen Liu, Bharath Ramsundar, Prasad Kawthekar, Jade Shi, Joseph Gomes, Quang Luu Nguyen, Stephen Ho, Jack Sloane, Paul Wender, and Vijay Pande. Retrosynthetic reaction prediction using neural sequence-to-sequence models. *ACS Central Science*, 3(10):1103–1113, 2017.

Qi Liu, Miltiadis Allamanis, Marc Brockschmidt, and Alexander Gaunt. Constrained graph variational autoencoders for molecule design. *Advances in Neural Information Processing Systems*, 31, 2018.

D Lowe. Chemical reactions from US patents (1976-Sep2016), 2017.

Daniel Mark Lowe. *Extraction of chemical structures and reactions from the literature*. PhD thesis, University of Cambridge, 2012.

Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *Artificial Neural Networks and Machine Learning–ICANN 2011: 21st International Conference on Artificial Neural Networks, Espoo, Finland, June 14-17, 2011, Proceedings, Part I 21*, pp. 52–59. Springer, 2011.

Tomoyuki Miyao, Hiromasa Kaneko, and Kimito Funatsu. Inverse QSPR/QSAR analysis for chemical structure generation (from y to x). *Journal of Chemical Information and Modeling*, 56(2):286–299, 2016.

Harry L Morgan. The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service. *Journal of Chemical Documentation*, 5(2):107–113, 1965.

Prakash M Nadkarni, Lucila Ohno-Machado, and Wendy W Chapman. Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, 18(5):544–551, 2011.

Dai Hai Nguyen and Koji Tsuda. Generating reaction trees with cascaded variational autoencoders. *The Journal of Chemical Physics*, 2022.

Mitsuru Ohno, Yoshihiro Hayashi, Qi Zhang, Yu Kaneko, and Ryo Yoshida. SMiPoly: Generation of synthesizable polymer virtual library using rule-based polymerization reactions. *Journal of Chemical Information and Modeling*, 2023 in press.

Quentin Perron, Olivier Mirguet, Hamza Tajmouati, Adam Skiredj, Anne Rojas, Arnaud Gohier, Pierre Ducrot, Marie-Pierre Bourguignon, Patricia Sansilvestri-Morel, Nicolas Do Huu, et al. Deep generative models for ligand-based de novo design applied to multi-parametric optimization. *Journal of Computational Chemistry*, 43 (10):692–703, 2022.

Giorgio Pesciullesi, Philippe Schwaller, Teodoro Laino, and Jean-Louis Reymond. Transfer learning enables the molecular transformer to predict regio-and stereoselective reactions on carbohydrates. *Nature Communications*, 11(1):1–8, 2020.

Munish Puri, Aum Solanki, Timothy Padawer, Srinivas M Tipparaju, Wilfrido Alejandro Moreno, and Yashwant Pathak. Introduction to artificial neural network (ANN) as a predictive tool for drug design, discovery, delivery, and disposition: Basic concepts and modeling. In *Artificial Neural Network for Drug Design, Delivery and Disposition*, pp. 3–13. Elsevier, 2016.

Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1(1):1–7, 2014.

Rampi Ramprasad, Rohit Batra, Ghanshyam Pilania, Arun Mannodi-Kanakkithodi, and Chiho Kim. Machine learning in materials informatics: recent applications and prospects. *npj Computational Materials*, 3(1):1–13, 2017.

Kunal Roy, Supratik Kar, and Rudra Narayan Das. Understanding the basics of QSAR for applications in pharmaceutical sciences and risk assessment. 2015.

Donald B Rubin. Using the SIR algorithm to simulate posterior distributions. *Bayesian Statistics*, 3:395–402, 1988.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

Mayu Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, pp. 4–11, 2014.

Benjamin Sanchez-Lengeling and Alán Aspuru-Guzik. Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, 361(6400): 360–365, 2018.

Ryusuke Sawada, Masaaki Kotera, and Yoshihiro Yamanishi. Benchmarking a wide range of chemical descriptors for drug-target interaction prediction using a chemogenomic approach. *Molecular Informatics*, 33(11-12):719–731, 2014.

Robert E Schapire and Yoav Freund. Boosting: Foundations and algorithms. *Kybernetes*, 2013.

Philippe Schwaller, Teodoro Laino, Théophile Gaudin, Peter Bolgar, Christopher A Hunter, Costas Bekas, and Alpha A Lee. Molecular transformer: a model for uncertainty-calibrated chemical reaction prediction. *ACS Central Science*, 5(9): 1572–1583, 2019.

Ari Seff, Wenda Zhou, Farhan Damani, Abigail Doyle, and Ryan P Adams. Discrete object generation with reversible inductive construction. In *Advances in Neural Information Processing Systems*, pp. 10353–10363, 2019.

Marwin HS Segler, Thierry Kogej, Christian Tyrchan, and Mark P Waller. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Central Science*, 4(1):120–131, 2018a.

Marwin HS Segler, Mike Preuss, and Mark P Waller. Planning chemical syntheses with deep neural networks and symbolic AI. *Nature*, 555(7698):604–610, 2018b.

Gregor Simm, Robert Pinsler, and José Miguel Hernández-Lobato. Reinforcement learning for molecular design guided by quantum mechanics. In *International Conference on Machine Learning*, pp. 8959–8969. Proceedings of Machine Learning Research, 2020.

Gregor NC Simm and José Miguel Hernández-Lobato. A generative model for molecular distance geometry. *arXiv preprint arXiv:1909.11459*, 2019.

Martin Simonovsky and Nikos Komodakis. Graphvae: Towards generation of small graphs using variational autoencoders. In *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part I 27*, pp. 412–422. Springer, 2018.

Masato Sumita, Kei Terayama, Naoya Suzuki, Shinsuke Ishihara, Ryo Tamura, Mandeep K Chahal, Daniel T Payne, Kazuki Yoshizoe, and Koji Tsuda. De novo creation of a naked eye–detectable fluorescent molecule based on quantum chemical computation and machine learning. *Science Advances*, 8(10):eabj3906, 2022.

David Weininger. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36, 1988.

Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

Stephen Wu, Yukiko Kondo, Masa-aki Kakimoto, Bin Yang, Hironao Yamada, Isao Kuwajima, Guillaume Lambard, Kenta Hongo, Yibin Xu, Junichiro Shiomi, et al. Machine-learning-assisted discovery of polymers with high thermal conductivity using a molecular design algorithm. *npj Computational Materials*, 5(1):1–11, 2019.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pp. 2048–2057. PMLR, 2015.

Xiufeng Yang, Jinzhe Zhang, Kazuki Yoshizoe, Kei Terayama, and Koji Tsuda. ChemTS: an efficient python library for de novo molecular generation. *Science and Technology of Advanced Materials*, 18(1):972–976, 2017.

Jiaxuan You, Bowen Liu, Zhitao Ying, Vijay Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. In *Advances in Neural Information Processing Systems*, pp. 6410–6421, 2018.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pp. 818–833. Springer, 2014.

Qi Zhang. Seq-Stack-Reaction. 2022. URL https://github.com/qi-zh/Seq-Stack-Reaction.

Qi Zhang, Chang Liu, Stephen Wu, Yoshihiro Hayashi, and Ryo Yoshida. A Bayesian method for concurrently designing molecules and synthetic reaction networks. *Science and Technology of Advanced Materials: Methods*, 3(1):2204994, 2023.

Shuangjia Zheng, Jiahua Rao, Zhongyue Zhang, Jun Xu, and Yuedong Yang. Predicting retrosynthetic reactions using self-corrected transformer neural networks. *Journal of Chemical Information and Modeling*, 60(1):47–55, 2019.