공학박사 학위논문

Deep Learning-Aided Autonomous Materials Discovery: Goal-Directed Inverse Molecular Design

딥러닝을 활용한 자율적 물질 발굴: 목표지향적 분자 역설계

2023년 2월

서울대학교 대학원

화학생물공학부

김 현 승

Deep Learning-Aided Autonomous Materials Discovery: Goal-Directed Inverse Molecular Design

딥러닝을 활용한 자율적 물질 발굴: 목표지향적 분자 역설계

지도 교수 이 원 보

이 논문을 공학박사 학위논문으로 제출함
2023년 2월

서울대학교 대학원
화학생물공학부
김 현 승

김현승의 공학박사 학위논문을 인준함
2023년 2월

위 원 장 이 규 태 (인)

부위원장 이 원 보 (인)

위 원 남 재 욱 (인)

위 원 신 동 일 (인)

위 원 나 종 걸 (인)

Abstract

Deep Learning-Aided Autonomous Materials Discovery: Goal-Directed Inverse Molecular Design

Hyunseung Kim
School of Chemical and Biological Engineering
Graduate School of Seoul National University

As the discovery of bronze marked the end of the Stone Age, the advancement of human civilization is closely related to the discovery of better materials. Modern materials discovery spans various fields ranging from electrical and electronic materials, energetic materials, ceramics, catalysts, nanomaterials, and biomaterials. In these kinds of materials development, it is challenging to search for the desired materials efficiently and quickly. In the past, searching for the desired materials relied on expert knowledge or intuition. However, this is not effective to search the vast chemical space efficiently and quickly. Therefore, attempts have been made to integrate the materials development process as a closed-loop system and drive it with AI/ML. The closed-loop materials discovery system driven by AI/ML consists of inter-module interactions such as inverse materials design, materials scoring, reaction pathway synthesis, and process design. Among the modules listed above, this study covers the module on inverse materials design. In this thesis, two kinds of

models have been addressed. One is a model for a goal-directed inverse molecular design using chemical language, which discovers realistic and chemically feasible molecular structure that hits a set of target properties. It is a model that embeds Transformer—a state-of-the-art natural language processing model—in a conditional variational autoencoder. It designs realistic and chemically feasible molecules by recognizing the patterns of linguistic sequence representing molecular structure. The other is a goal-directed inverse molecular design based on AI-driven combinatorial chemistry, which enables materials discovery with extreme properties; note that existing probability-distribution learning models such as neural machine translator, generative adversarial network, and variational autoencoder-based inverse molecular design models cannot generate molecules with such rare and extreme properties out of known materials (training data) distribution. The original combinatorial chemistry is a method that generates molecules from the combination of randomly selected molecular fragments. Hence, this method can generate materials with all possible properties that can be obtained from the combination of molecular fragments, even materials with extreme properties. However, it lacks the policy of selecting molecular fragments to hit the target properties. For this reason, the proposed method uses reinforcement learning to learn the policy of selecting molecular fragments to guide it to the desired target. Considering the models work their tasks accurately in sub-seconds, it is believed that deep learning-aided models will contribute to accelerating the materials development process.

Keywords: materials discovery, goal-directed inverse molecular design, chemical language, fragment-based RL, AI-driven combinatorial chemistry

Student Number: 2018-20184

Table of Contents

List of Tables

List of Figures

Chapter 1. Introduction

1.1. Objectives and scope

This thesis aims to contribute to accelerating the material development process through AI/ML-aided materials discovery research. Accelerating the materials development process is attracting a lot of attention since it can reduce the cost of materials development in the material industry and help preoccupy intellectual property rights and the market.

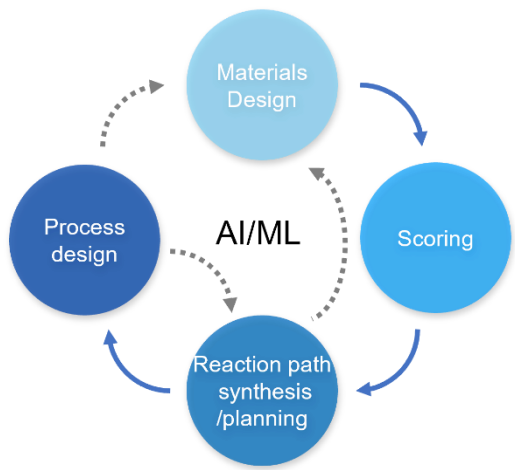


Figure 1 A closed-loop system for AI/ML-driven materials discovery.

Efforts to accelerate the process of materials development began in earnest when the practical use of big data began to increase along with the rapid advancement of hardware. With big-data technology, data-based material development projects have begun since the term material genome was first used in 2002. **[1]** In particular, the material genome initiative launched in the United States in 2011 aimed to cut the material development period by half, thereby innovatively

reducing the material development period. In addition, the rapid development of AI/ML in the past decade has provided an atmosphere that can further accelerate materials discovery; the European materials modeling council, launched in 2016, aims to reduce the material development period to minutes. **[2]**

The closed-loop material discovery system driven by AI/ML will help to find the desired material accurately and quickly without human intervention. **[3]** It effectively reduces the cost required for material development. A closed-loop system for product and process discovery driven by AI/ML can be composed of the interaction of four modules (**Figure 1**): the inverse materials design module, the materials scoring module, the reaction path synthesis/planning module, and the process design module. Such a closed-loop materials discovery system is expected to contribute greatly to future materials discovery research as AI/ML autonomously and automatically handles the decision-making.

Existing material development research has used a method of evaluating a set of potential target material candidates through experiments and computer-aided simulations and screening the materials that meet the given conditions step by step. Since there was no methodology capable of accurately and rapidly designing molecular structures (desired output) from target properties or functionality (given inputs), the screening method has been the best to use. However, recent advances in deep learning (DL) and machine learning (ML) have made it possible to derive models that can effectively infer the desired output from given input via trained patterns of inputs and outputs represented in empirical data. This makes it possible to develop inverse molecular design models that can directly infer the hit-like materials from the given task. This not only effectively reduces the number of materials to be evaluated, but also effectively increases the potential of the material

candidates to be tested.

The goal-directed inverse materials design module can generate potential target material candidates, but it does not inform which of the generated materials is closest to the desired use. Note that even with goal-directed materials design module, the generated molecules are not exactly matched to the set of target properties. Therefore, a materials scoring module is needed to sort the generated molecules in the order of fit for a given task. The reason why such a sorting process is necessary is that the costs of subsequent processes are high, e.g., searching feasible reaction paths to generate the designed product from commercially available raw materials, and process design for it.

In this thesis, two major problems are addressed for goal-directed inverse materials design. One relates to the inverse design of natural molecules that hits a set of multi-objective properties and/or functionalities. The other is the problem of inverse molecular design to be used even when there are no known molecular samples that hit the set of given target properties or are not enough to train the artificial neural network models. This problem is related to materials discovery that founds better materials with extreme properties and/or functionalities which have never been observed. All the models listed above are expected to enable autonomous material discovery driven by AI/ML by combining these with the additional modules of materials scoring, reaction path synthesis, and the process construction module.

1.2. Thesis organization

- Chapter 2. Molecular representation methods:

  In this chapter, methods of representing molecules as quantitative data are

introduced, and the advantages and disadvantages of each representation method are summarized.

- Chapter 3. Materials design with context-awareness:

  This chapter describes a problem of natural molecules design (like real molecules) that hits multi-objective properties. In this problem, after training the chemical language representing molecules through the language model embedded in the conditional generative model (namely, generative chemical Transformer, GCT **[4]**), hit-like & realistic materials are generated with the understanding of the patterns of the corpus (referring the patterns of molecular substructures) of the chemical compounds. The attention-mechanism inherent in the GCT helps to generate molecules that conform to the chemical valence rule by recognizing the structural information of molecules hidden within the chemical language.

- Chapter 4. Materials design with extreme properties:

  This chapter describes the problem of discovering materials with extreme properties out of known materials' distribution. The problem is that it is hard to gauss the molecular structure of potential molecules from no similar samples or few known samples. It is also a problem for the probability-distribution learning models (e.g., translator and generative model-based models) which are dominantly used to inverse molecular design. These models that learn the empirical data-distribution are not suitable for inferring materials outside the trained-data-distribution. In contrast, combinatorial chemistry can generate all materials with properties that can

be obtained from random combinations of predefined molecular fragments. However, combinatorial chemistry has a problem in that it has no policy to select fragments to combine, which is needed to derive hit-like materials. For this reason, reinforcement learning (RL) is introduced to combinatorial chemistry to give a fragment selection policy to it.

- Chapter 5. Closing remarks:

  This chapter states the conclusions of this thesis. It summarizes the contribution of this thesis to AI/ML-aided materials discovery and discusses the challenges and opportunities.

Chapter 2. Molecular representation methods

2.1. Tribes of molecular representation methods

To learn molecular data in artificial neural networks, it is necessary to represent molecular data in the form of structured data. There are many tribes to represent molecules e.g., fingerprints, molecular graph, chemical language, potentials, coulomb matrix, bag of bonds, fragments, 3D geometry, and electronic density. Here, fingerprints, molecular graph, and chemical language are the most popular and widely used. In this chapter, the pros and cons of these representation methods are summarized (**Table 1**).

The fingerprint is a method that represents molecules as a hash function. It has a predefined dictionary constituting a molecular fragment list. Extended-Connectivity Fingerprints **[5]** and Morgan Fingerprints **[6]** are popularly used. The fingerprint representations have the advantages of fixed-size representation. It represents various-sized molecules as the data of fixed-sized vectors. In addition, if the molecular fragment list is set by considering the reaction path, it can be used to design materials that have a reaction path to produce them.

Another molecular representation method is the molecular graph. It represents molecules as node features and edge features. The node and edge correspond to the atom and bond, respectively. Hence, the nature of molecules resembles graph representation. However, it is not easy to use to design materials since the node features and edge features are highly correlated. For example, to convert a molecular graph into a molecule, the chemical valence rule must be satisfied. To do this, the

combination of node features and edge features must be feasible. However, data generated by generative models may contain artifacts, and the molecular graph data with a single defect cannot be converted into a normal molecule. For this reason, using the graph representation method in molecular generation tasks, well-designed constraints should be needed. Another characteristic of graph representation is permutation variation. Even for the same molecule, different representations occur depending on the order in which each atom or bond is written. In other words, various representations are possible according to the order in which each atom or bond is written. This can be both an advantage and a disadvantage. For problems with small data, it has the advantage of amplifying data by representing permutation differently. On the other hand, there are various problems caused by the expression of one molecule not being matched one-to-one.

The other method is chemical language representation, which represents a three-dimensional molecular structure as a one-dimensional sequence. It is easy to store and retrieve data by representing complex molecular structures in a one-dimensional sequence. In particular, since natural language processing models actively being studied in the field of deep learning are based on sequence data, there is an advantage in that advanced natural language processing models can be applied to chemical language learning. However, as the size of the molecule increases, an information gap may occur in representing the three-dimensional structure in a one-dimensional sequence. Furthermore, this method also has the same characteristic of graph representation, a permutation variation.

Table 1 Molecular representation methods

Methods		
Fingerprints	Pros	- Hash Function: Molecules can be represented as vectors of predefined size, regardless of molecular size - Molecules can be fragmented or combined with specific rules; this can be applied to reaction pathway synthesis
	Pros & Cons	- During inverse design, the exact molecular structure cannot be specified
Molecular graph	Pros	- The molecular structure itself and the representation are most similar
	Pros & Cons	- Depending on the permutation of vertices and edges, there are various representations for the same molecule
	Cons	- Generation of intact molecular graph data without constraint rules is difficult
Chemical language	Pros	- Easy to store and utilize molecular data - High-level natural language models are available; context recognition ability can be highly utilized
	Pros & Cons	- Depending on the permutation of vertices and edges, there are various representations for the same molecule
	Cons	- As the molecular size increases, information loss may occur in representing the three-dimensional molecular structure as a one-dimensional representation

2.2. Which is the best representation method for deep learning?

As stated in the previous section, there are many ways to represent molecules as structured data. Each has advantages and disadvantages, and it is hard to specify which method is the best at present. One of the reasons is that it has not been long since this field has been actively researched. **Figure 2** summarizes the publication years of papers found by searching for the keyword 'deep learning + materials' on Scopus (accessed on Nov. 16, 2022). In my opinion, since it was March 2016 when Deep Mind's AlphaGo won the Go against the best Go player, Lee Se-dol, studies that applied deep learning to the material field began to be actively reported in 2017 or 2018. Therefore, it seems that it is still too early to reach a consensus on which methods are most appropriate for each task. Another reason is that it is difficult to ascertain whether the difference in performance between models is due to a difference in representation methods (e.g., graph, language, and fingerprints) or a difference in the performance of the model itself. For example, even though a model using a graph-based methodology showed higher performance than a model using a chemical language-based methodology, it cannot be generalized that the graph-based methodology is better than the language-based methodology. For this reason, research that presents metrics that can quantitatively evaluate the performance of each model and a benchmarking platform composed of them is very important for the development of this field. **[7]** Note that Prof. Vijay Pande's group presents the latest benchmarking results with various models for the fields of quantum mechanics, physical chemistry, biophysics, and physiology. **[8]**

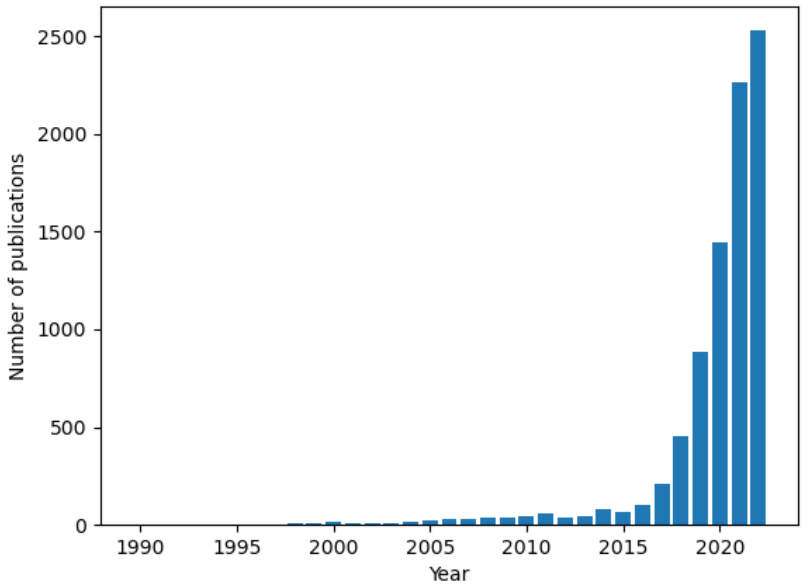


Figure 2 The publication years of papers found by searching for the keyword 'deep learning + materials' on Scopus (accessed on Nov. 16, 2022).

Chapter 3. Materials design with context-awareness

3.1. Introduction

Materials discovery is a field of research that finds materials that are suitable for the desired use. The hit-like materials must satisfy some given target properties or functionalities. Traditionally, the discovery was conducted by evaluating the properties of the candidates through computational simulations and experiments for obtained candidate materials to find the best material that hit the given target properties (forward materials design). High-throughput screening is a typical example. This kind of approach still occupies the mainstream of materials discovery, but there are also problems. For example, it could be difficult in securing a group of potential candidates that are likely to contain the hitting materials for the given targets. In addition, there is a problem of high cost in the evaluation of numerous candidates. On the other hand, goal-directed inverse materials design can solve problems by directly designing materials that are likely to hit the given target properties or functionality at the same time. For this kind of approach, it is difficult to derive a generalized function (white-box model) using a theoretical formula to specify the molecular structure from the input target properties or functionalities. Hence, a black-box model such as deep-learning can be a good solution to this kind of problem where a clear mechanism for determining potential outputs from an input is not known. This is because deep learning can derive an approximation function that infers the desired outputs from a given input by learning the relationship between input and output hidden in empirical data.

To learn the molecular structures in artificial neural nets, the molecular

structures must be represented as structured data. Molecules are made up of atoms and connecting bonds. The molecules can be represented as graphs constituting nodes and edges. Also, line notation can be used to represent the molecular structure, namely, chemical language. The chemical language represents a molecule with a line notated sequence with various symbols indicating the type of atom and type of bond. It also has grammar like natural language. This kind of chemical language was introduced to efficiently record and retrieve the structural information of molecules. **[9-13]** Simplified molecular-input line-entry system (SMILES) **[11-13]**—developed in the 1980s—is the most popular chemical language.

Several approaches have demonstrated that Natural Language Processing (NLP) models are applicable to inverse molecular design problems. The NLP models design molecules by generating chemical language. It generates chemical language by iteratively selecting a character that follows the currently generated string. **[14-18]** Some studies proposed a language model combined with a variational autoencoder (VAE) **[19]**. It represents molecular information into compressed latent space and generates chemical language from resampled latent code. **[20,21]** Some others proposed a language model combined with a generative adversarial network (GAN) **[22]**. It also generated chemical language from sampled noise. **[23]** Reinforcement learning (RL) is also applicable to this problem to obtain a policy to select a character constituting a chemical language. **[24]**

An important point in materials discovery is to find hit-like materials that meet multiple desired target properties. The desired molecules can be discovered in two steps by applying additional optimization or navigation process to the generative model: Bayesian optimization, **[20,25]** particle swarm optimization, **[26]** genetic

optimization, **[27-29]** or Monte Carlo tree search. **[30-32]** In the first step, it generates random molecules. Then select the best molecule in the second step. Another kind of method is to use conditional models which generate molecules with the given target conditions in a single step. It can shorten the time consuming to discover the desired molecules and directly control the molecules to be generated by manipulating the input conditions (goal-directed inverse materials design). A conditional recurrent neural network (cRNN) **[33]** is a neural machine translator (NMT) that generates desired molecules by translating input target conditions to corresponding chemical language. Unlike generative models, it is limited to generating various molecular candidates with a single condition set. This is because the recurrent neural network (RNN) is fundamentally a one-to-one matching translator, not a generative model; here, the generative models refer to models that can output various results by decoding noise (actually, latent code).

In this chapter, generative chemical Transformer (GCT), **[4]** which embeds Transformer **[34]**—a state-of-the-art architecture that became a breakthrough for NLP problems by using an attention mechanism **[35]**—into a conditional variational autoencoder (cVAE) is proposed. From the point of view of data recognition and processing, GCT is close to a conditional generator that embeds the Transformer's language recognition ability based on the attention mechanism. It is designed to take advantage of both the high-performance language model and the conditional generative model. Trained GCT is analyzed by quantitatively evaluating the generated molecules. GCT shows many strong points. When analyzing the attention-score calculated by GCT, it is confirmed that a high attention-score is given between correlated parts in molecular structure. Chemical languages generated by GCT were

more likely to satisfy chemical rules than those generated by other models used for benchmarking. It is believed that a deep understanding of the molecular structure by paying attention to each character in chemical strings sparsely helps the generation process (grammar understanding). In addition, the chemical strings are parsed into highly realistic molecules (context-awareness). And also, it is demonstrated that the conditional variational generator, which is the skeleton of GCT, helps to generate molecules that satisfy multiple given conditions simultaneously (conditional generator) and varies for a single set of conditions (variational generator). In addition, the autoencoder, a substructure of GCT, makes the molecular size controllable.

3.2. Backgrounds

3.2.1 SMILES Representation

SMILES, a chemical language, represents molecules as one-dimensional text. It is a powerful language since a one-SMILES string is converted into an exact molecule. It represents the atoms, bonds, and rings that makeup molecules as a string. An atom is represented by the alphabet of the element symbol (e.g., C, N, O, and H), and a bond is represented by a single bond (-), double bonds (=), and triple bonds (#), depending on the type. In general, a bond that can be easily inferred through the atoms or ring structure of the surrounding atom is omitted. The notation of hydrogen is also omitted in the SMILES string if single-bond hydrogen can be explicitly inferred by the chemical valence rule, however, single-bond hydrogen can be indicated by using [H] if the bond is implicit. For charged atoms, where the number of hydrogen bonds cannot be determined explicitly, atoms and formal charges are written together in brackets [ ]. The beginning and end of each ring are represented as the same digit, and the pair must be correct; if a ring is open, it must be closed. The atoms present in the aromatic ring are written in lowercase, while the atoms outside the ring are capitalized. The branches in molecules are indicated by opening and closing parentheses (see **Figure 3**b). A more detailed description of SMILES is in the **ref.** [4-6].

3.2.2 Limitation of Language Representation

Unfortunately, language representation has a limitation since most molecules

are non-Hamiltonian graphs. Here, the Hamiltonian graph refers to a graph that can derive a path that passes through all edges in a graph only once. And non-Hamiltonian graph refers to a graph that cannot derive the path. It is self-evident that semantic discontinuity occurs whenever a branch in a molecule is translated into a one-dimensional string (**Figure 3**a). SMILES distinguishes each branch with open and close parentheses and creates a gap between the distance of two characters within the string and the distance of the corresponding two atoms in the molecular graph. An example is shown in **Figure 3**b. Even though atom2 and atom13 in **Figure 3**b are neighbors in the molecular graph, they are placed far from each other in the string. For this reason, the longer the branch is, the harder it is to imagine the molecular structure by reading the chemical string in sequence order (similar to memory cells). A similar phenomenon happens in natural language (**Figure 3**c). The longer the sentence is, the larger the gap between the characters' placement and the semantic similarity. Unfortunately, in chemical language, there are more areas where semantic discontinuity occurs. In the field of NLP, by introducing an attention mechanism to this problem, language models find correlated parts beyond the placement of words by paying attention to semantically related parts; it is a departure from the traditional way of perceiving context in the order of sentences (memory cells). The Transformer is an architecture involving an attention mechanism in the form of a neural network, and it became a breakthrough in context-aware ability. This study started with the expectation that the attention mechanism could help structural understanding beyond the semantic discontinuity of chemical language.

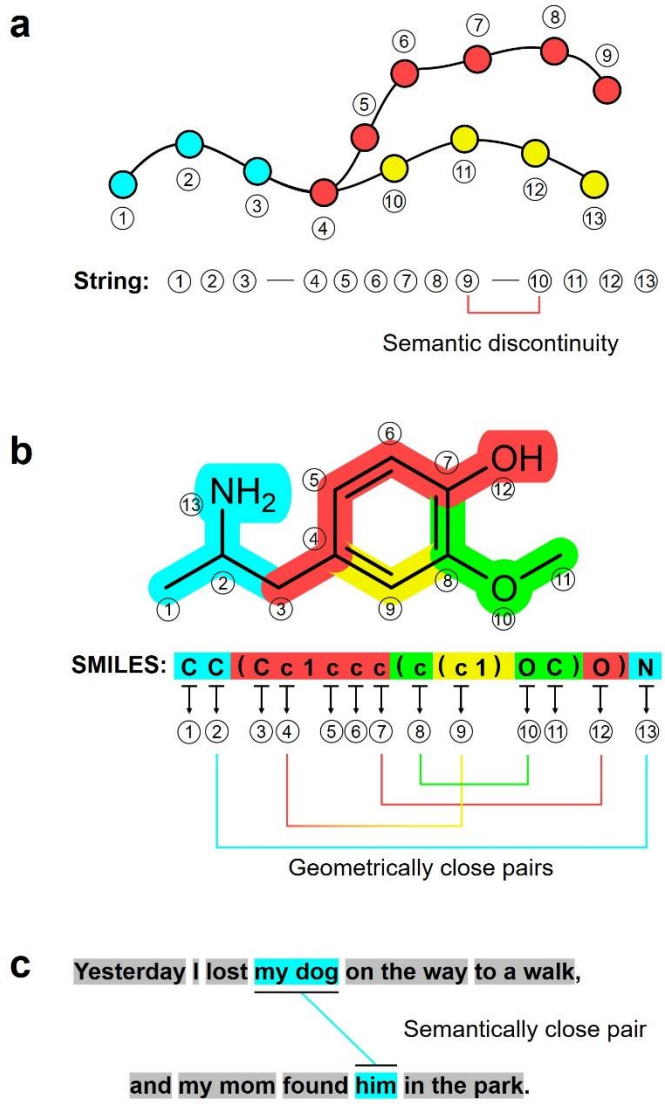


Figure 3 Structural limitation of language forms. a An example of a non-Hamiltonian graph. A Hamiltonian graph has a path that passes through all the points in the graph only once, and a non-Hamiltonian graph does not have such a path. **b** An example of a non-Hamiltonian molecular graph and its SMILES string: 4-(2-aminopropyl)-2-methoxyphenol. Each atom is labelled with a circled number. Different colors refer to different branches. **c** In natural language, words that are semantically close within a sentence are not always structurally close within a sentence. Reprinted with permission from **ref. [4]**. Copyright 2021 American Chemical Society.

3.3 Methods

3.3.1 Tokenization and Token Embedding

To input SMILES string to language models, the process of tokenizing by semantic units is necessary. The SmilesPE[29] tokenizer was used to tokenize the SMILES strings included in the training data of Molecular Sets benchmarking platforms (MOSES).30 In total, 28 types of tokens are used: 4 special tokens (<unknown>, <pad>, <sos>, and <eos>), 13 atom tokens (<C>, <c>, <O>, <o>, <N>, <n>, <F>, <S>, <s>, <Cl>,
, <[nH]>, and <[H]>), 3 bond tokens (<->, <=>, and <#>), 2 branch tokens (<(> and <)>), and 6 ring tokens (<1>, <2>, <3>, <4>, <5>, and <6>). Note that tokens related to charged atoms (e.g. <[O-]>, <[n+]>) and tokens related to stereochemistry (e.g. </> , <\>) were not considered as they are not covered by the MOSES database. Each token that constitutes the SMILES string is one-hot encoded in 28 dimensions and embedded in 512 dimensions. The condition of GCT is also embedded in 512 dimensions.

3.3.2 Attention Mechanism

The attention mechanism is the core of the Transformer's language cognition abilities. The attention mechanism allows the Transformer to self-learn which token of the input string is better to focus on to perform a given task better. The attention mechanism uses three vectors: the query Q, the key K, and the value V. The attention mechanism calculates the similarity between Q and all keys in K, and the

calculated similarity is multiplied by the value corresponding to the key to calculate the attention scores. The scale-dot attention used in the Transformer is calculated as follows **[27]**:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1)$$

where d_k is the dimension of K and d_k must correspond to the dimension d_q of Q. The Transformer uses multi-head attention instead of single-head attention (**eq. 2**) **[34]**:

$$\begin{gathered}\text{MultiHead}(Q, K, V) = \text{Concat}(head_1, \ldots, head_h)W^O \\ where\ head_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)\end{gathered} \quad (2)$$

where $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^O \in \mathbb{R}^{hd_v \times d_{model}}$ and $d_{model} = d_k \times h = d_v \times h$ Here, h is the number of multi-head and d_v is the dimension of V. Each head ($head_i$) calculates an attention score between Q and K from different viewpoints using the different weights belonging to each head (W_i^Q, W_i^K, and W_i^V).

3.3.3 Generative Chemical Transformer

GCT, an architecture that embeds Transformer—one of the most advanced language models—into a cVAE **[37]** which generates SMILES hitting target properties based on a deep understanding of chemical language is proposed. Transformer, the core of GCT's language recognition ability, is mainly used as an

NMT. It consists of an encoder and a decoder (**Figure 4**a). The encoder receives a sentence to be translated and understands the received sentence through self-attention. Then, the processed information from sentence comprehension is passed to the decoder. The decoder iteratively selects the next token that will follow the translated sentence up to this point, referring to the information received from the encoder and the sentences translated up to the previous step; if there is no translated sentence at the beginning of translation, the special token 'start of sentence <sos>' is used. The decoder uses the input information to iteratively select the next token that will follow the translated sentence up to the previous step. Finally, the translation ends when the decoder selects a special token 'end of sentence <eos>'.

GCT is a structure that inserts a low-dimensional conditional Gaussian latent space between the encoder and the decoder of the Pre-Layer Normalization (Pre-LN) Transformer. **[38]** (**Figure 4**b). Pre-LN Transformer is a modified version of the original (Post-Layer Normalization, Post-LN) Transformer. The combination of language models and variational autoencoders is vulnerable to posterior collapse. **[39]** A complete solution to posterior collapse has yet to be identified; however, it is known that Kullback-Leibler divergence (KL) annealing can alleviate this problem. **[40]** Since KL annealing (KLA) controls the gradient size, adopting Pre-LN Transformer—designed to stabilize the gradient flow of the (Post-LN) Transformer—can facilitate KLA manipulation. The loss function of GCT is as follows:

$$\begin{aligned} L(\emptyset, \theta; x_{enc}, x_{<t}, x_t, c) \\ &= k_w D_{KL}\Big(q_\emptyset(z|x_{enc}, c) \parallel p(z|c)\Big) \\ &- E_{q_\emptyset(z|x_{enc}, c)}[\log g_\theta(x_t|z, x_{<t}, c)] \end{aligned} \tag{2}$$

where $D\_KL(\cdot)$ is the KL divergence, and $E[\cdot]$ is the expectation. $q_\emptyset$ is a parameterized encoder function, $g_\emptyset$ is a parameterized decoder function (generator), $p(\cdot|c)$ is a conditional Gaussian prior. Here, $\emptyset$, θ, x_{enc}, $x_{<t}$, z, x_t, c, k_w are the parameter set of the encoder, the parameter set of the decoder, the input of the encoder, the input of the decoder, the latent variables, the reconstruction target, the conditions, and the weight for KLA, respectively. The encoder and decoder each consist of six Pre-LN Transformer blocks. Each block has dimensions of 512 and 8-head attention, and the dimension of the feed-forward block is 2,048. The Gaussian latent space is designed in 128 dimensions.

The self-attention block of the encoder obtains the concatenated array of the SMILES string and three different properties: the octanol-water partition coefficient (logP), **[41]** the topological polar surface area (TPSA), **[42]** and the quantitative estimate of drug-likeness (QED) **[43]**. The encoder-decoder attention block in the decoder obtains the concatenated array of latent code and condition (three properties), and the self-attention block in the decoder obtains only the SMILES string. In the training phase, GCT performs the task of reconstructing the SMILES string—input through the encoder—by referring to the given hints on the molecular properties. In this process, the low-dimensional latent space acts as the model's bottleneck to find as much meaningful information that can be restored to the decoder as possible by exploiting the limited information passed through the bottleneck. Then, meaningful latent variables for molecular structures and properties are represented in the low-dimensional continuous latent space. In the inference phase, a sampled latent code and target properties are input into the learned decoder, and the decoder selects the next tokens iteratively through a 4-beam search **[44]** (a kind of tree search method).

A dropout **[45]** rate of GCT is 0.3 applied. Learning is conducted by the Adam optimizer **[46]**. The initial learning rate is 10^{-4}. The expansion rate of the momentum is 0.9 and the expansion rate of the adaptive term is 0.98. Two methods are applied to schedule the learning rate (GCT-WarmUP and GCT-SGDR in **Table 2**). One is to use the warm-up scheduler (**eq. 3**) **[34]**:

$$\eta = 3\, d_{model}^{-0.5} \cdot \min\left(s_{current}{}^{-0.5}, s_{warmup}{}^{-1.5}\right) \quad (3)$$

where η means learning rate, $s_{current}$ means the current training step, and s_{warmup} means warm-up steps. s_{warmup} is set to 100,000. The other is to use stochastic gradient descent with warm restart (SGDR) of one epoch cycle (**eq. 4**) **[47]**:

$$\eta = \eta_{min} + 0.5(\eta_{max} - \eta_{min})\left(1 + cos\left(\frac{s_{current}}{s_{cycle}}\pi\right)\right) \quad (4)$$

where, η_{min} means minimum learning rate, η_{max} means maximum learning rate, s_{cycle} is learning rate scheduling step cycle. Here, η_{min}, η_{max}, and s_{cycle} are set to 0, 0.0001, and one-epoch steps, respectively. KL annealing was applied to increase k_w from 0.02 to 0.50 at 0.02 intervals per epoch for a total of 25 epochs.

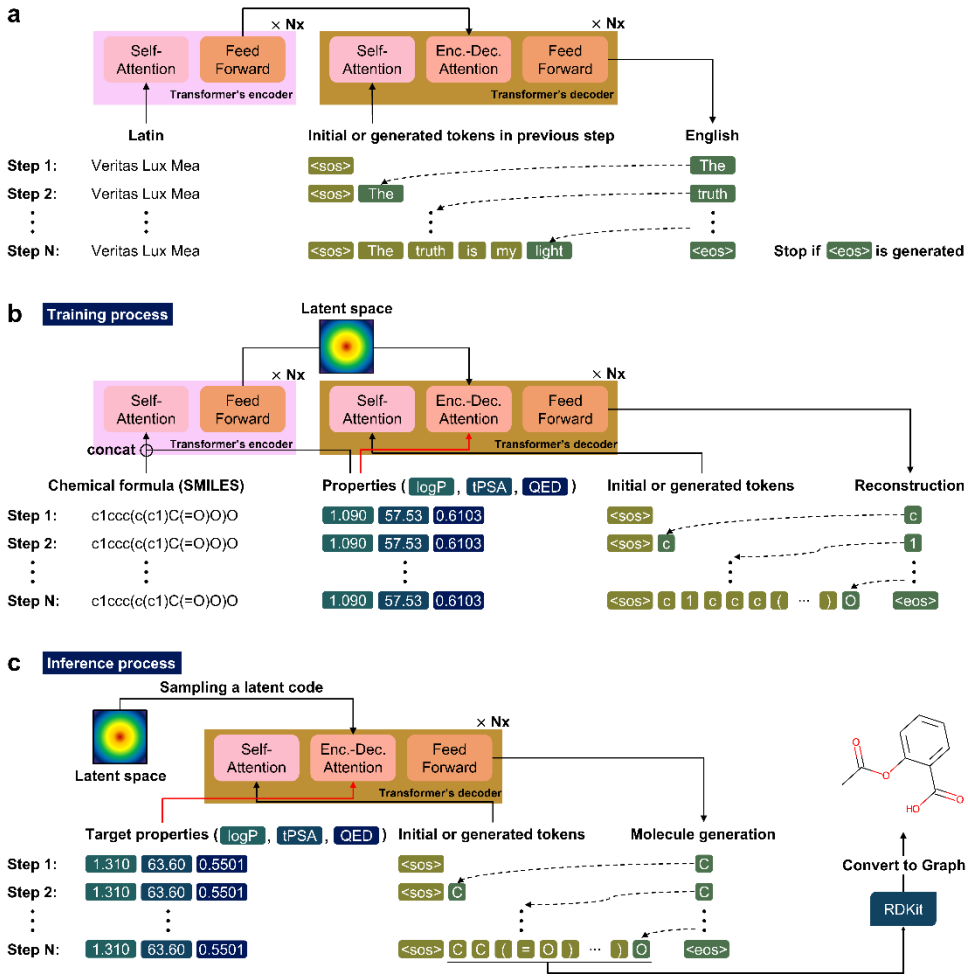


Figure 4 De novo molecular generation using GCT. a Transformer model for NMT: an example of translating Latin into English. It iteratively selects the next English word by referring to the Latin sentence and the English sentence translated up to the previous step. **b** In the process of learning to reproduce the input chemical formula, GCT learns the molecular structure and three different properties: logP, TPSA, and QED. It represents the information on molecular structure and properties in the latent space during the learning process. **c** The trained GCT generates a de novo molecule that satisfies the target properties by decoding the molecular information sampled from the latent space and the given preconditions. Reprinted with permission from **ref. [4]**. Copyright 2021 American Chemical Society.

3.3.4 Datasets & Benchmarking Metrics

The GCT model was trained and benchmarked using a database of MOSES benchmarking platforms. The MOSES database is a benchmarking dataset for drug discovery created by sampling molecules from the ZINC **[48]** is Not Commercial (ZINC) database—composed of commercially available compounds—that satisfy specific conditions: molecular weight in the range from 250 to 350 daltons, number of rotatable bonds is not greater than 7, not containing charged atoms or atoms other than C, N, S, O, F, Cl, Br, and H or cycles longer than eight atoms. The MOSES database consists of training samples (1.7 M), test samples (176 k), and scaffold test samples (176 k), which have scaffolds that never appear in the training samples. It is also designed to closely match the distribution between the datasets. The three additional properties (logP, TPSA, and QED) computed from RDKit **[49]** are used for GCT learning.

In general, the quality of network training can be evaluated by measuring how different the model's predicted and the actual labels are. However, for molecular generative models, the small mean loss does not guarantee that the generative model performs well because the artifacts in the generated molecules, which are not observed in the mean loss measurement, may not fit the chemical valence rule or may make the molecules unrealistic. For this reason, the quality of the generated molecules needs to be checked against the following criteria:

- How plausible are the generated molecules?
- Do the generated molecules satisfy the target properties?

- Can multiple candidates be generated for a single precondition set?

- Can de novo molecules be created in a short time?

In total, 30,000 SMILES strings are generated by the trained GCT model and evaluated by MOSES benchmarking score metrics (**Table 2**). In addition to relatively simple scores such as validity, uniqueness, internal diversity, filters, and novelty, the MOSES benchmarking platform also provides metrics that can measure similarity with reference molecules such as the Similarity to a Nearest Neighbor (SNN), **[7]** Fréchet ChemNet Distance (FCD), **[50]** Fragment similarity (Frag), **[7]** and Scaffold similarity (Scaf). **[7]**

The SNN score is calculated as follows:

$$\mathrm{SNN}(G,R) = \frac{1}{|G|}\sum_{m_G \in G} \max_{m_R \in R} T(m_G, m_R) \tag{5}$$

where G and R refer to the set of molecules generated and reference molecules, respectively, m stands for Morgan fingerprints, **[8]** and $T(A,B)$ stands for the Tanimoto similarity **[51]** between set A and set B. SNN is a metric to evaluate the similarity to the nearest neighbor $m_R \in R$, which has the highest Tanimoto similarity for the molecule m_G to be evaluated. Thus, the more similar two molecules m_G and m_R are, the higher the SNN score is.

The FCD uses activation of the penultimate layer in ChemNet and is designed to predict bioactivity. It calculates the difference in the distributions between G and R is calculated as follows:

$$\mathrm{FCD}(G,R) = \|\mu_G - \mu_R\|^2 + Tr\left(\sum\nolimits_G + \sum\nolimits_R - 2\left(\sum\nolimits_G \sum\nolimits_R\right)^{1/2}\right) \quad (6)$$

where μ is the mean, $\sum$ is the covariance, and $Tr(\cdot)$ is the trace operator. The more similar the two sets of G and R, the lower the FCD value.

The Frag score is calculated as follows:

$$\mathrm{Frag}(G,R) = \frac{\sum_{f\in F}\left(c_f(G)\cdot c_f(R)\right)}{\sqrt{\sum_{f\in F} c_f^2(G)}\sqrt{\sum_{f\in F} c_f^2(R)}} \quad (7)$$

where F is the set of 58,315 unique BRICS fragments, **[52]** and $C_f(A)$ is the frequency with which fragment $f \in F$ appears in the molecules in set A. It is a metric that measures the similarity of the frequency in which each fragment belonging to BRICS appears in both sets of G and R. Thus, the more similar the distributions of the two sets are, the higher the Frag score is.

The Scaf score is calculated as follows:

$$\mathrm{Scaf}(G,R) = \frac{\sum_{s\in S}(c_s(G)\cdot c_s(R))}{\sqrt{\sum_{s\in S} c_s^2(G)}\sqrt{\sum_{s\in S} c_s^2(R)}} \quad (8)$$

where S is the set of 448,854 unique Bemis-Murcko scaffolds **[53]** and $C_s(A)$ is the frequency at which scaffold $s \in S$ appears in the molecules in set A. The scaffold set S used for evaluation is curated by MOSES. The more similar the distributions of the two sets are, the higher the Scaf score is.

3.3.5 Condition Sampling

The properties considered in this problem are logP, TPSA, and QED. A three-dimensional histogram was derived after dividing each property into 1,000 equal

sections between the maximum and minimum values in the training data. Then, the cells were sampled according to the probability that data samples exist in each cell; here, the probability is the number of samples in that cell out of the total samples. Next, uniform noise was added at the center value of the cell to create condition sets for the 30,000 molecules to be generated; the sizes of the uniform noise for the logP-axis, TPSA-axis, and QED-axis are applied to not exceed the size of the cell sides in each axis direction.

3.3.6 Latent Variable Sampling

As mentioned earlier, the dimension of latent variables is set to 128; 128-number of latent variables. However, since the number of tokens constituting a SMILES string is various for each molecule, the sequence length of the latent variables is applied differently each time; $\mathbb{R}^{128\times sequence\_length}$. Here, the sequence length means the number of tokens constituting the SMILES string. The sequence length used for each molecular generation was sampled from a normal distribution. The mean and variance of the normal distribution were derived from the number of tokens constituting the SMILES strings in the MOSES training dataset. After the sequence length is determined, the values of the latent variables are sampled from the standard normal distribution.

3.4 Results and Discussion

3.4.1 How Plausible Are the Generated Molecules?

To evaluate whether the generated SMILES strings represent plausible molecular structures, analysis from two perspectives is required. The first analysis is whether the generated SMILES strings can generate valid molecular graphs, in other words, whether the generated SMILES strings satisfy both the chemical valence rule and the syntax of the SMILES language. From the benchmarking results, it was found that more than 98.5% of the generated SMILES strings are valid; GCT-WarmUp shows a validity of 98.5% and GCT-SGDR shows the validity of 99.2%. It is the highest value among language-based models (**Table 2**a). This ability depends on how well the generative machine can understand the geometry of molecules through SMILES strings. To determine the character (corresponding to atom, bond, or branch) followed by the given chemical string that satisfies the chemical valence rule and the grammar of the SMILES language, the geometry of the molecules (connectivity of each atom and the branches) present in the string must be understood. It seems that the attention mechanism applied to GCT helps the neural network to understand the grammar of chemical language beyond the semantic discontinuity of the SMILES language. It tends to be consistent with the results (visualized example of attention score for diproxadol) shown in **ref. [54]**.

Figure 5 shows the results for two extreme examples of how to pay attention to the characters within the SMILES string. Atom1, atom2, and atom13 in **Figure 5**a are located close to each other in the molecular graphs but far away from each other within the SMILES string. Although only a SMILES string was provided to GCT, it

is recognized that atom1, atom2, and atom13 are related to each other (♠); **Figure 5**b shows that GCT-SGDR recognizes the relationship between atom2 and atom22, 23 (♦). It also recognizes atoms corresponding to a particular branch (♣) and recognizes the ring type of branch (♥). Each attention-head recognizes the molecular structure according to different viewpoints. In summary, the attention mechanism applied to GCT seems to help GCT to recognize the molecular structure hidden in the one-dimensional text. This claim is consistent with the claim of **ref. [54]** regarding the role of the attention mechanism.

In addition, it was confirmed that the GCT showed the highest SNN score among the compared models for the indicators (SNN, FCD, Frag, Scaf) that measure the similarity to the real molecules. However, FCD, Frag, and Scaf were not as good as the evaluation results of other compared models. Note that among the compared similarity metrics, SNN has a characteristic that distinguishes it from other metrics. As can be seen from **eq. 5-8**, FCD, Frag, and Scaf are metrics that compare the distribution between the generated molecular set and the reference molecule set (real molecular set). On the other hand, SNN is a metric that measures the mutual similarity between a generated molecule to be evaluated and a real molecule having the highest similarity (nearest neighbor). Hence, SNN is a metric independent of the distribution between the two sets. Note that artificial neural networks are universal approximators that approximate empirical probability distributions. Therefore, the distribution of the generated data approximates the distribution of the trained data very well. However, in the case of GCT, since the distribution of the generated molecular population may be manipulated according to given input conditions, the distribution of the generated molecular population may deviate from the distribution

of the trained molecular population. Since the test set and TestSF are intentionally designed to be similar to the distribution of the training set, it is believed that higher FCD, Frag, and Scaf scores were obtained in the unconditional models used for comparison. However, GCT showed the highest score for the SNN score, which is evaluated by one-to-one comparison rather than the comparison of two sets' distribution. For further verification, the structures of molecules generated by GCT and real molecules were compared (**Figure 6**). As can be seen from the figure, it was confirmed that the molecules generated by GCT contained substructures similar to those of real molecules.

Table 2 Comparison of the MOSES benchmarking. Adapted with permission from **ref. [4]**. Copyright 2021 American Chemical Society.

			GCT (This works)		MOSES Reference Models				
			GCT -WarmUp	GCT -SGDR	VAE	AAE	Char RNN	Latent GAN	JTN -VAE **[55]**
Validity	↑		0.9853	**0.9916**	0.9767 ±0.0012	0.9368± 0.0341	0.9748 ±0.0264	0.8966 ±0.0029	1.0±0.0
Unique@1k	↑		**1.0**	0.998	**1.0±0.0**	**1.0±0.0**	**1.0±0.0**	**1.0±0.0**	**1.0±0.0**
Unique@10k	↑		0.9981	0.9797	0.9984 ±0.0005	0.9973 ±0.002	**0.9994 ±0.0003**	0.9968 ±0.0002	0.9996 ±0.0003
Novelty	↑		0.8144	0.6756	0.6949 ±0.0069	0.7931 ±0.0285	0.8419 ±0.0509	**0.9498 ±0.0006**	0.9143 ±0.0058
SNN	↑	Test	0.6179	**0.6513**	0.6257 ±0.0005	0.6081 ±0.0043	0.6015 ±0.0206	0.5371 ±0.0004	0.5477 ±0.0076
		TestSF	0.5771	**0.5990**	0.5783 ±0.0008	0.5677 ±0.0045	0.5649 ±0.0142	0.5132 ±0.0002	0.5194 ±0.007
FCD	↓	Test	0.4017	0.7980	0.0990 ±0.0125	0.5555 ±0.2033	**0.0732 ±0.0247**	0.2968 ±0.0087	0.3954 ±0.0234
		TestSF	0.8031	0.9949	0.5670 ±0.0338	1.0572 ±0.2375	**0.5204 ±0.0379**	0.8281 ±0.0117	0.9382 ±0.0531
Frag	↑	Test	0.9973	0.9922	0.9994 ±0.0001	0.9910 ±0.0051	**0.9998 ±0.0002**	0.9986 ±0.0004	0.9965 ±0.0003
		TestSF	0.9952	0.8562	**0.9984 ±0.0003**	0.9905 ±0.0039	0.9983 ±0.0003	0.9972 ±0.0007	0.9947 ±0.0002
Scaf	↑	Test	0.8905	0.8562	**0.9386 ±0.0021**	0.9022 ±0.0375	0.9242 ±0.0058	0.8867 ±0.0009	0.8964 ±0.0039
		TestSF	0.0921	0.0551	0.0588 ±0.0095	0.0789 ±0.009	**0.1101 ±0.0081**	0.1072 ±0.0098	0.1009 ±0.0105

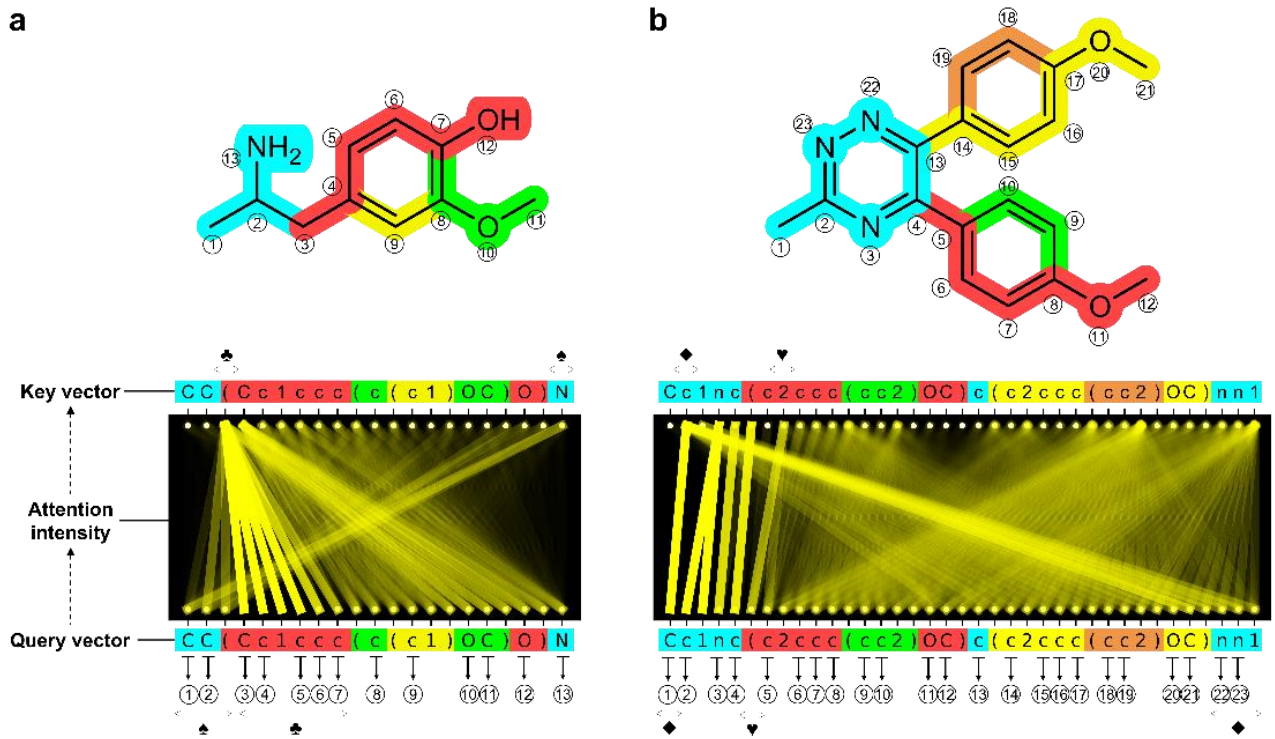


Figure 5 Visualized examples for self-attention scores of an attention head in encoder blocks. a Visualization results of the fourth head in the second encoder block for 4-(2-aminopropyl)-2-methoxyphenol. **b** Visualization results of the third head in the second encoder block for 5,6-bis(p-methoxyphenyl)-3-methyl-1,2,4-triazine. The visualization scheme is borrowed from **ref. [56]**. Reprinted with permission from **ref. [4]**. Copyright 2021 American Chemical Society.

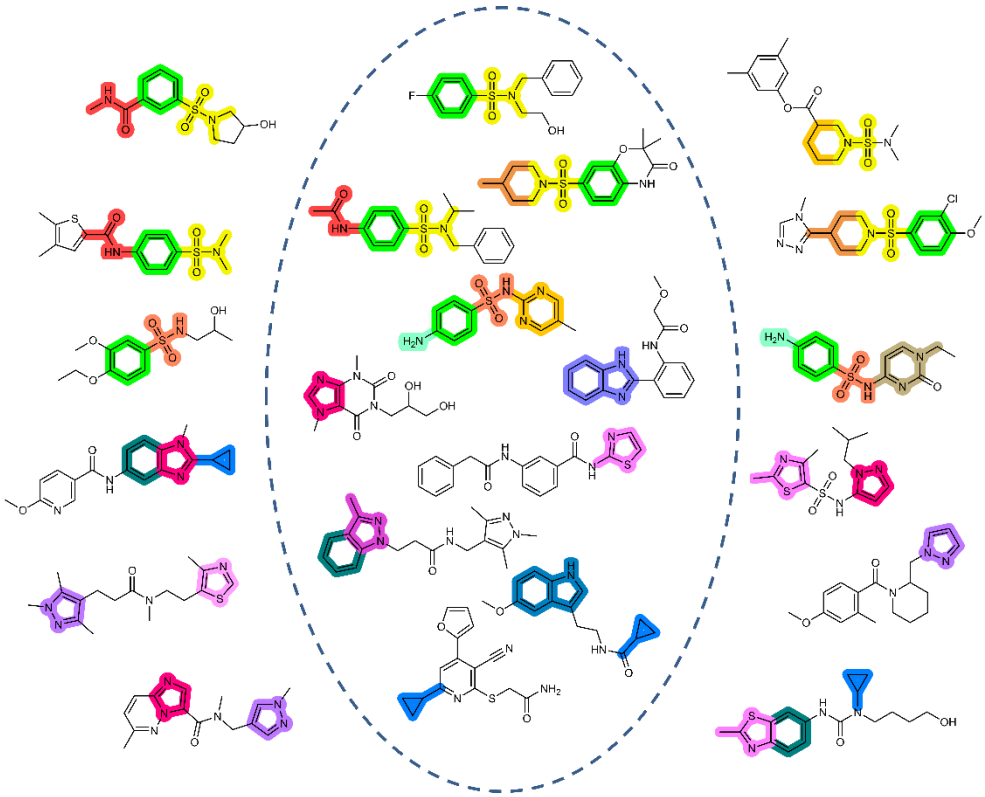


Figure 6 Comparison between the generated molecules and real molecules. The molecules in the dotted line are the real molecules in the MOSES training set, and the molecules outside are the generated molecules.

3.4.2 Do the Generated Molecules Hit the Target Properties?

GCT generates molecular structures that hit multiple target properties. **Figure 7**a-c are the results of comparing the properties of 30,000 molecules generated from GCT (calculated from RDKit) and target properties (preconditions given in GCT). Since logP and TPSA are physical properties directly related to the molecular structure, it is possible to generate a molecular structure corresponding to the target property based on an understanding of the molecular structure. However, the QED is an artificial index designed to determine the likeness to drugs quantitatively through geometric averages of eight different properties, so it is relatively difficult for the QED; this phenomenon is also found with cRNNs. The absolute mean errors between the target conditions for each property and the properties of the generated molecule are 0.177 (logP), 2.923 (TPSA), and 0.035 (QED).

The length of the generated SMILES string depends on the length of the latent code since GCT has an autoencoder (AE) structure; it is trained to reconstruct information input into the encoder. In the training phase, the length of the latent code appears equal to the length of the string input into the encoder and the length of the string output from the decoder; in fact, these are slightly different depending on whether the <sos> and <eos> tokens are used in the input and output design. In the inference phase, the length of the input latent code and the length of the generated SMILES string did not match perfectly and GCT does not learn the distribution of sequence lengths (**Figure 7**d). However, it seems that the length of the generated SMILES string which is related to the size of the molecule can be manipulated to some extent by adjusting the length of the latent code.

To check whether the multiple given target properties are satisfied simultaneously, the properties of generated molecules were compared to the 10 precondition sets that were sampled from the distribution of training data (**Figure 7**e-g). The conditional model, which is a skeleton of GCT, generates molecules that simultaneously satisfy multiple target properties well. Furthermore, the variational generator in GCT makes it possible to generate various molecules under the same precondition set (**Figure S1-10**).

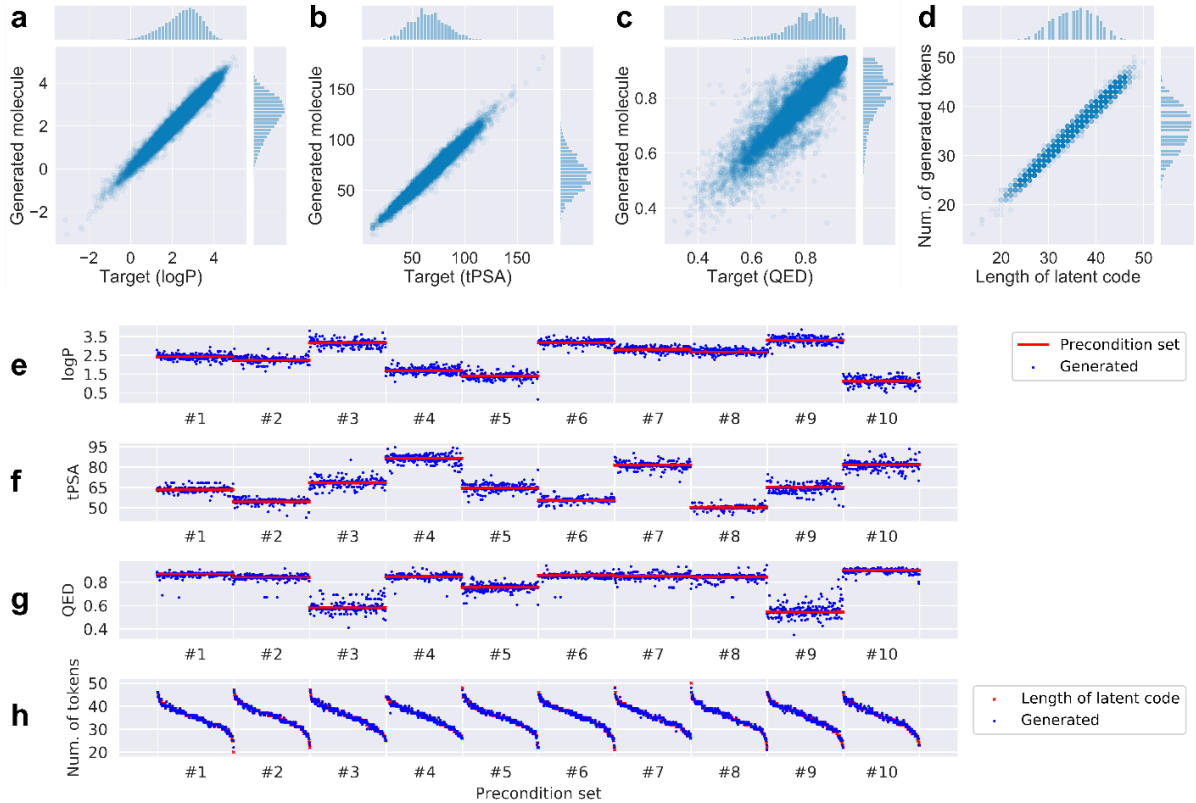


Figure 7 Evaluation of target hitting ability. a-d Parity plots between the target properties and the properties of 30,000 generated molecules. **e-h** Parity plots between target properties (red line) and properties of generated molecules for each set of targets (blue dots). 10,000 generation trials were conducted for each set of targets. Reprinted with permission from **ref. [4]**. Copyright 2021 American Chemical Society.

3.4.3 Can De Novo Molecules Be Created?

Whether a generative model can create de novo molecules is an important criterion that determines its applicability for material discovery. The novelty score refers to the probability of generating a new molecule that does not exist in the training data (**Table 2**). Note that only a high novelty score does not guarantee that it is a good generator since odd and unrealistic molecules can increase the novelty score. Hence, the novelty score should be used in conjunction with indicators to evaluate whether the generated molecules are realistic. **Figure 8** shows a scatter plot of each model's novelty score and validity score. The dotted line is a linear regression of the reference model scores (VAE, AAE, CharRNN, GAN, JTN-VAE). Interestingly, for all reference models, it is observed that the novelty score decreases as the SNN score increases. Conversely, the higher the novelty score, the lower the SNN score. This means that it is not easy to create new molecules that are similar to real molecules. However, it can be confirmed that GCTs (GCT-WarmUp, GCT-SGDR, GCT-Exp1~3) generate new and similar molecules better than the reference models; Here, GCTs mean trained models using different hyperparameters or different learning rate schedules. The detailed conditions and scores for each GCT are summarized in **Table 3**.

The model using the SGDR (GCT-SGDR) shows lower novelty and higher validity than the models using the warm-up scheduler (GCT-WarmUp). A scheduler that cyclically reduces the learning rate has a loss in reducing the KL divergence term of the VAE loss function, however, it has a benefit in reducing the reconstruction error term. **[57]** It seems that the SGDR scheduler, a kind of cyclic

annealing scheduler, makes GCT-SGCR have high validity and low novelty. The time taken per molecule generation was 507 ms in the environment of an 8C/16T CPU and an NVIDIA GTX 1080 Ti and 440 ms with a 12C/24T and an NVIDIA Tesla T4.

3.4.4 Limitations

Not all properties of molecules used for drug discovery were considered, and only properties of drug molecules covered by the MOSES dataset were considered; charged atoms and stereochemistry are not considered and these are limitations of this study. Furthermore, it is hard to extrapolate outside of the property window GCT was trained on since VAE, which is a model that learns the distribution of data and generates data by sampling the latent variables from the learned distribution of latent variables, cannot learn the distribution of data properly for regions where there are no data samples or for sparse regions. For this reason, the target properties are not satisfied relatively well for precondition #10 in **Figure 7**e-h; the low QED area has few data samples (see **Figure 7**c).

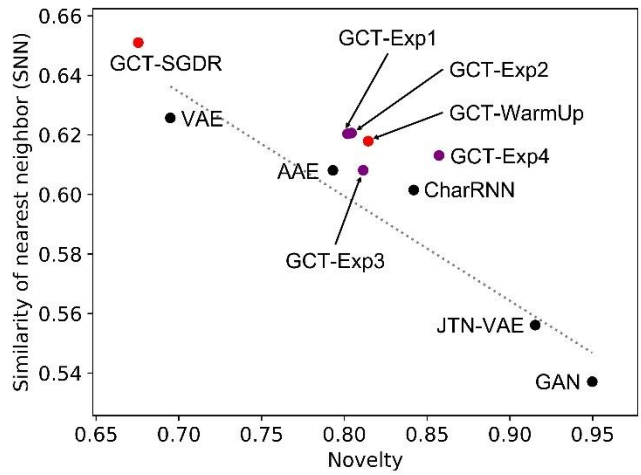


Figure 8 Novelty-SNN relationship. Reprinted with permission from **ref. [4]**. Copyright 2021 American Chemical Society.

Table 3 MOSES benchmarking results with various training conditions set-up. Reprinted with permission from **ref. [4]**. Copyright 2021 American Chemical Society.

	GCT-Exp1 (baseline)	GCT -WarmUp	GCT -Exp2	GCT -Exp3	GCT -Exp4	GCT -SGDR
Training Conditions						
Epochs	25	25	25	25	**50**	25
KLA weight (k_w) [start:step:end]	[0:0.02:0.5]	[0:0.02:0.5]	[0:0.02:0.5]	**[0:0.01:0.25]**	**[0:0.01:0.5]**	[0:0.02:0.5]
Learning rate (lr)	Baseline lr	**Baseline lr × 3 (= eq. 3)**	**Baseline lr × 5**	Baseline lr	Baseline lr	**eq. 4**
Learning rate schedular	Warm-Up	Warm-Up	Warm-Up	Warm-Up	Warm-Up	**SGDR**
MOSES Benchmarks						
Validity	0.9757	**0.9853**	0.9808	0.9692	0.9813	**0.9916**
Unique@1k	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**0.998**
Unique@10k	0.9977	**0.9981**	0.9977	0.9983	**0.9986**	**0.9797**
Novelty	0.8019	**0.8144**	0.8043	0.8114	**0.8572**	**0.6756**
SNN (Test)	0.6204	**0.6179**	0.6206	0.6081	0.6131	**0.6513**
SNN (TestSF)	0.5784	**0.5771**	0.5791	0.5693	0.5744	**0.5990**
FCD (Test)	0.4181	**0.4017**	**0.3813**	0.3883	0.7180	**0.7980**
FCD (TestSF)	0.8560	**0.8031**	0.8411	**0.7855**	1.221	**0.9949**
Frag (Test)	0.9977	**0.9973**	**0.9980**	0.9979	0.9944	**0.9922**
Frag (TestSF)	**0.9957**	**0.9952**	0.9956	0.9960	0.9910	**0.8562**
Scaf (Test)	**0.9021**	**0.8905**	0.8935	0.8644	0.8893	**0.8562**
Scaf (TestSF)	0.0883	**0.0921**	0.1039	**0.1049**	0.0969	**0.0551**

3.5 Conclusion

In this chapter, a GCT architecture that embeds Transformer—a language model that has been a breakthrough in the field of NLP using an attention mechanism—into a conditional variational generator was studied. The trained GCT can generate SMILES strings that meet the desired conditions based on a deep understanding of chemical language. It learns molecular structures and three different properties as a form of language: the logP, TPSA, and QED. Quantitative evaluations have been performed by scoring molecules converted from the generated SMILES strings. In this process, the characteristics of the metrics (the SNN, FCD, Frag, and Scaf) that measure the plausibility of the molecules were analyzed, and the limitations were discussed. The performance of GCT has been benchmarked by the MOSES benchmarking platform. By analyzing the results, it has been demonstrated that GCT can utilize both the advantages of a language model and a conditional variational generator. The conclusions obtained are summarized as follows:

- (1) The attention mechanism in GCT helps to deeply understand the molecular structures beyond the limitations of chemical language semantic discontinuity resulting from converting a non-Hamiltonian molecular graph to a one-dimensional string by paying sparse attention to chemical formulas.

- A deep understanding of chemical language makes the generated SMILES strings (2) satisfy the syntax of SMILES language, (3) satisfy the chemical rules, and (4) are realistic.

- The conditional variational generator in GCT makes the generated molecules (5) satisfy multiple target properties simultaneously and (6) vary.

- The AE structure of GCT (7) makes the molecular size controllable.

- GCT (8) creates de novo molecules that have never been seen in the training process, and (9) creates a molecule in hundreds of milliseconds.

Well-trained GCT (GCT-WarmUp) generates valid SMILES strings with 98.5% probability. 84.1% of the generated SMILES strings were new molecules that had never been learned, and their similarity with real molecules was 0.681 (their SNN score was 0.681). It is difficult to create a new molecule that is similar to the pattern of existing real molecules even never been seen before. Among the compared models, GCT showed the best performance in making new molecules that are similar to real molecules. Additionally, the generated molecules satisfied multiple target properties simultaneously, and the mean absolute errors for the three different properties were 0.177 (logP), 2.923 (TPSA), and 0.035 (QED). In addition, it has been confirmed that GCT can control the molecular size; the averaged difference in the number of generated SMILES tokens compared to the given length of latent code is 0.332. Molecular generation took 507 ms per molecule on a personal computer. Furthermore, conditions applicable to GCT can be adjusted. It is believed that GCT can be extended to Transformer-based architectures such as BERT, **[58]** GPT, **[59]** and T5 **[60]**. Considering the time required versus the advantages listed above, it is expected that our proposed model can contribute to accelerating the process of materials discovery.

Chapter 4. Materials design with extreme properties

4.1 Introduction

Practical material discovery tasks often require the discovery of materials with rare properties rather than those commonly present. For example, polymers with better mechanical properties or drugs with better activity. These are related to materials discovery with extreme properties or functionality, which are superior to those of the already discovered. **[61-63]** In another case, there is a need for materials in which two or more properties or functionalities are difficult to appear together. **[64]** For example, engineering plastics used in automobiles need to be light while the mechanical strength is as strong as metal. **[64-66]** The problem is that these materials are relatively rare in nature. It means that there aren't many known molecular structures for reference. If many examples of molecular structures with properties similar to the desired properties are known, the molecular structure of the desired materials can be inferred from their common characteristics and expert knowledge. However, in the opposite cases, it is difficult to infer the desired molecular structure from nothing or insufficiently few known samples. Unfortunately, it is true not only for human experts but also for models learning the data.

As for the inverse molecular design, data probability distribution learning models based on NMT or generative models such as VAE and GAN have been mainly used. All these are models that approximate the probability distribution of training data. In the case of using NMT-based models, these design molecules inversely by translating input target properties into their molecular structures. These

are kinds of a method that learn the relationship between the properties and molecular structures. In the cases of using VAE or GAN-based models, these generate molecules from sampled noise. The generators are trained to approximate the probability distribution of data to be generated by the generator to the distribution of training data. **Figure 9** compares the distribution of the training data provided by the MOSES benchmarking platform and the molecules inferred by the models trained on it. As can be seen from the figure, the generated data is similarly distributed to its training data. It means that the probability distribution learning models approximate the distribution of training data very well. Therefore, paradoxically, these probability distribution learning models do not guarantee that these models will operate as intended for the region out of trained data-distribution. In other words, it may not be suitable to use these models for the problems of discovering materials out of training data distribution, which is to be solved in this chapter.

On the other hand, combinatorial chemistry **[67]** can infer various materials that deviate from the distribution of known material samples. Combinatorial chemistry is a methodology for generating molecules by combining predefined molecular fragments with rules for combining them. Therefore, even for the target properties that have never been discovered, combinatorial chemistry can generate materials that hit the target if the target is included in the set of obtainable properties form the combination of predefined fragments. Hence, it would be more suitable for the problem of materials discovery with extreme properties. However, combinatorial chemistry has a critical limitation on the fragment selection policy; it randomly selects molecular fragments to be combined. In other words, there is no policy of molecular fragment selection to guide the compound to the desired target. This

means that it entails a problem of combinatorial explosion. It can be an obstacle to quickly and effectively discovering the desired material.

In this chapter, a model for learning the molecular fragment selection policy that leads combinatorial chemistry to derive hit-like materials through RL has been proposed (fragment-based RL). Through this, the proposed model can utilize the strengths of combinatorial chemistry (the ability to discover materials with extreme properties), while solving the disadvantages of combinatorial chemistry (the absence of a fragment selection policy). Studies that attempt inverse molecular design using fragment-based RL are not entirely new. **[68-70]** However, to the best of the author's knowledge, no study insists on the use of fragment-based RL for materials discovery with extreme properties. This study contributes to three major points. Firstly, it is theoretically reviewed that the probability distribution learning models such as NMT and generative models (e.g., VAEs and GANs) do not guarantee the materials discovery out of trained data distribution. Secondly, it has been empirically confirmed that the probability-distribution learning models cannot discover materials out of trained data distribution, however, fragment-based RL can do it. Thirdly, it has been confirmed that the fragment-based RL is a universal solution to various materials discovery with extreme properties by conducting several case studies: multi-objective hitting materials discovery and lead molecules discovery to a protein target.

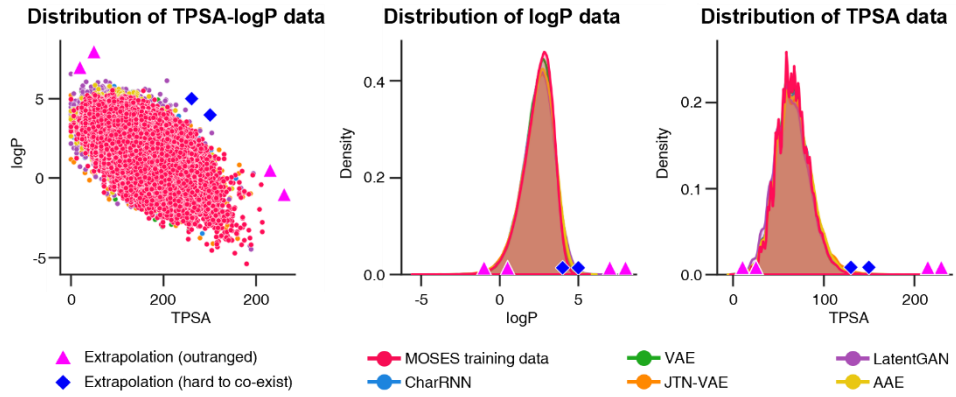


Figure 9 Distribution of MOSES training data and data generated by MOSES baseline models that learn MOSES training data.

4.2 Theoretical Review of Probability-Distribution Learning Models

For inverse molecular design problems, sequential classification models based on NMT or generative models based on VAE or GAN are mainly used. All these are models that learn the empirical probability-distribution of training data. In this section, it is theoretically shown that minimizing the loss functions of the models mentioned above is equal to an approximation of the probability distribution of training data with its generator.

Let the training data including N-samples and their labels be $X, Y = (X_1, Y_1), \dots, (X_N, Y_N)$ where each $X_i = \left(x_{i,1}, \dots, x_{i,T}\right)$ is a chemical language consisting of T one-hot encoded tokens x_i, and $Y_i = \left(y_{i,1}, \dots, y_{i,T'}\right)$ are T' labeled properties corresponding to X_i. First, let the NMT model $G_\theta^{NMT}(X|Y)$ be a θ-parameterized model that translates the input target property Y into the chemical language X. NMT calculates θ^* that maximizes the maximum likelihood for the training data to derive the optimal model $G_{\theta^*}^{NLP}(X|Y)$. This is equivalent to deriving θ^* that minimizes the negative log-likelihood $-\log G_{\theta^*}^{NMT}(X|Y)$ for the training data. It is equal to minimizing the expectation of negative log-likelihood for all data samples present in training data sets: $-\sum_i^N P(X_i|Y_i) \log P\left(G_\theta^{NMT}(X_i|Y_i)\right) = -\sum_i^N P(X_i|Y_i) \log P_\theta\left(\hat{X}_i \middle| Y_i\right)$ where $\hat{X}_i$ is the hypothesis. Therefore, the negative log-likelihood loss is equal to the cross entropy $H\left((X, Y), \left(\hat{X}, Y\right)\right)$, which can be rewritten in terms of entropy $((X, Y))$ and the KL-divergence term as follows:

$$
\begin{aligned}
H\left((X,Y),(\hat{X},Y)\right) &= \sum_{i}^{N} P(X_i|Y_i)\log P_\theta\left(\hat{X}_i|Y_i\right) \\
&= \sum_{i}^{N} P(X_i|Y_i)\log P_\theta(X_i|Y_i) \\
&\quad + P(X_i|Y_i)\log P_\theta\left(\hat{X}_i|Y_i\right) - P(X_i|Y_i)\log P_\theta(X_i|Y_i) \\
&= H((X,Y)) + D_{KL}\left(P(X|Y) \parallel P_\theta\left(\hat{X}|Y\right)\right)
\end{aligned} \tag{9}
$$

where $H((X,Y))$ denotes the entropy of the training data X,Y. Since the $H((X,Y))$ is a term independent of the trainable parameter θ, minimizing $H\left((X,Y),(\hat{X},Y)\right)$ is equivalent to minimizing $D_{KL}\left(P(X|Y) \parallel P_\theta(\hat{X}|Y)\right)$. Here, $P(X|Y)$ in the KL-divergence term is a fixed term from the training data. Therefore, by tuning the trainable parameter θ to make $P_\theta\left(\hat{X}|Y\right)$ closed to $P(X|Y)$, θ^* that minimizes $D_{KL}\left(\mathrm{P(X)} \parallel P_\theta(\hat{X})\right)$ is obtained. In other words, training the NMT model is equal to obtaining $G_{\theta^*}^{NMT}(X|Y)$ in the way mentioned above, which derives a generator that approximates empirical data distribution.

VAE is a model that generates the target X. It represents the training data X in the latent space using the $\emptyset$-parameterized encoder $Q_\emptyset^{VAE}$. Then, after concatenating the sampled latent code z and the target property Y, inputting it into G_θ^{VAE} to reconstruct X. In the case of VAE, the same conclusion can be drawn by transforming the loss function of VAE into cross-entropy. It can be rewritten in terms of entropy and the KL-divergence term. The loss function of VAE is as follows:

$$
\sum_{i}^{N} \mathbb{E}_{Q_\emptyset^{VAE}(z|X_i)}\left[\log G_\theta^{VAE}(X_i|z,Y_i)\right] + D_{KL}\left(Q_\emptyset^{VAE}(z|X_i) \parallel P(z)\right) \tag{10}
$$

where $Q_\emptyset^{VAE}$ and z mean $\emptyset$-parameterized encoder and latent code, respectively. In the training process, the KL-divergence term approximates the distribution of the

encoded latent code to the prior distribution $P(z)$ and acts like a regularizer. The expectation term acts like a reconstruction error. Looking at the reconstruction error from the point of view of the generative model, it can be understood as calculating the trainable parameters $\emptyset$ and θ that minimize the log-likelihood $\log G_\theta^{VAE}(X_i|z,Y_i)$ in the process of minimizing the loss. As mentioned earlier, log-likelihood is $\sum_i^N P(X_i|z,Y_i)\log P\left(G_\theta^{VAE}(X_i|z,Y_i)\right)$ with $z\sim Q_\emptyset^{VAE}(z|X_i)$, and this can be simply expressed as $\sum_i^N P(X_i|z,Y_i)\log P_\theta\left(\hat{X}_i\middle|z,Y_i\right)$ with $z\sim Q_\emptyset^{VAE}(z|X_i)$. This is equivalent to cross-entropy $H\left((X,z,Y),\left(\hat{X},z,Y\right)\right)$ which can be rewritten in terms of entropy $H((X,z,Y))$and the KL-divergence term:

$$\begin{aligned} H\left((X,z,Y),\left(\hat{X},z,Y\right)\right) &= \sum_i^N P(X_i|z,Y_i)\log P_\theta\left(\hat{X}_i\middle|z,Y_i\right) \\ &= \sum_i^N P(X_i|z,Y_i)\log P_\theta(X_i|z,Y_i) \\ &\quad + P(X_i|z,Y_i)\log P_\theta\left(\hat{X}_i\middle|z,Y_i\right) \\ &\quad - P(X_i|z,Y_i)\log P_\theta(X_i|z,Y_i) \\ &= H\left((X,z,Y)\right) + D_{KL}\left(P(X|z,Y)\parallel P_\theta\left(\hat{X}\middle|z,Y\right)\right) \end{aligned} \tag{11}$$

Here, minimizing $H\left((X,z,Y),\left(\hat{X},z,Y\right)\right)$ is the same as deriving $G_\theta^{VAE}(X_i|z,Y_i)$ that makes $P_\theta\left(\hat{X}\middle|z,Y\right)$ approximate $P(X|z,Y)$ with $z\sim Q_\emptyset^{VAE}(z|X_i)$. That is, G_θ^{VAE} is a model that approximates the empirical probability-distribution of the training data by tuning θ to make the probability distribution of data generated by G_θ^{VAE} closed to the probability distribution of training data.

In the case of GAN, G_θ^{GAN} has been trained to generate data with noise until the discriminator $D_\emptyset^{GAN}$ cannot distinguish the generated and the real correctly. Note that GAN has been proven in the **ref. [22]** to be a global minimum of the virtual training criterion of the generator is achieved if and only if $P(X)=G_\theta^{GAN}(z)$. That

is, the GAN is also a model that makes the distribution of the data generated by the generator approximate the empirical probability-distribution of the training data.

The problem is that the true probability of the considering system cannot be obtained. Therefore, it is impossible to check whether the empirical probability-distribution that the generator approximates is close to the true probability. Furthermore, in the case of materials discovery with extreme properties covered in this chapter, which is hard to get sufficient data samples, the distribution of the gathered data (training data) is likely to deviate from the true probability of the considering system. Therefore, it is hard to be guaranteed that the probability-distribution learning models based on NMT, VAE, and GAN work as intended in the problem of materials discovery with extreme properties.

4.3 Fragment-based RL

Fragment-based RL is a methodology that uses RL to learn the molecular fragment selection policy required for combinatorial chemistry. RL is a method that derives a mature policy through trial and error according to the reward derived from the results of the agent's action in the environment. So, it is not a method that learns the probability distribution of training data.

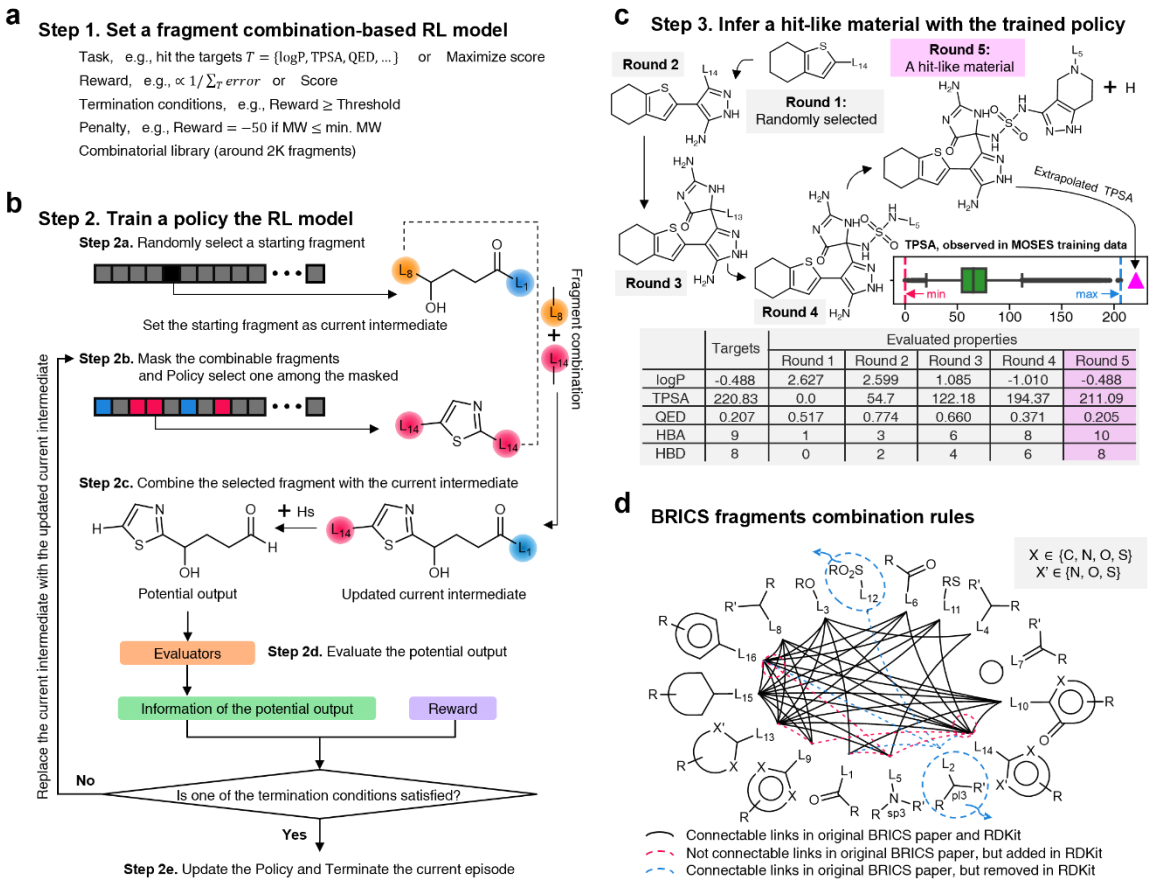
Before training the RL model, a basic configuration is required (**Figure 10**a). First, it is necessary to understand a given task. It is used to design a reward function suitable for the task. For example, in the case of a multi-objective hitting problem, the reward function can be an inverse form of the sum of errors for each objective. If the given task is to maximize or minimize some score, the score itself can be used

as a reward. That is, the reward function can be designed to output the higher the reward if the better the given task is performed. Next, a penalty function can be applied to prevent undesired results. For example, if only too small molecules are generated, the sufficient diversity of the discovered materials cannot be secured. To prevent this, a penalty may be given in the range where the molecular weight (MW) is too small. The next thing to do is to set the termination conditions. Termination conditions are the criteria for deciding whether to attach additional molecular fragments to the potential molecular structures or finish the game without further fragment selection. For example, not selecting any more fragments if the reward exceeds a threshold can be used as one of the termination conditions. If any one of the termination conditions is satisfied, the potential molecular structure completed so far is output as a final product. For reference, there are cases that the hit-like material has been found until the game is terminated, but there are also cases that the hit-like material cannot be found despite going through too many rounds of molecular fragment selection. In the latter case, not selecting any more fragments if the reward exceeds a given threshold can be one of the termination conditions. Here, a maximum MW could be one of the termination conditions. The next thing to do is to set up the combinatorial library, a set of molecular fragments to use. In this study, about 2K fragments were selected from the BRICS 4K fragment set provided by breaking of retrosynthetically interesting chemical substructures (BRICS) **[52]** and used as a combinatorial library. BRICS also provides rules for combining molecular fragments (**Figure 10**d). The original BRICS combination rules support combinations that can be combined among 16 templates (union set of solid lines and blue dotted lines). In this study, the modified BRICS combination rule provided by RDKit is adopted (union set of solid lines and red dotted lines). The modified BRICS

combination rule uses 14 templates. Each template is a generalized BRICS fragment. Each template represents multiple BRICS fragments. All templates have combinable site(s) ($L_{x:1:16}$). If there is a BRICS combination rule that two templates can be combined, two BRICS fragments which are corresponding to each template can be combined by attaching a binding site.

After the basic configuration is completed, the fragment selection policy will be trained (**Figure 10**b). In the training process, the initial fragment is randomly selected from among the predefined combinatorial library (Step 2a). This method is adopted to secure the versatility of the final product. The next step is masking fragments which can be combined with the current intermediate; if it is the first round for molecular fragment selection, the initial fragment is the current intermediate. Then, the policy selects a fragment among the masked fragments (Step 2b). It is a kind of action masking method to reduce the action space efficiently. The newly selected fragment is attached to the current intermediate. There are two cases of the state of the current intermediate. One is that the current intermediate has one or more unoccupied binding sites. In this case, a hydrogen atom(s) is attached to the unoccupied binding site according to the chemical valence rule. This process is necessary to make the current intermediate intact since only intact molecules can be evaluated. The other is that the current intermediate has no unoccupied binding site. In this case, the current intermediate is intact. So, it does not need any attachment to evaluate it (Step 2c). The intact molecule is named potential output. The potential output is evaluated with the evaluator(s) (Step 2d). Then, the game is terminated and update the policy with an obtained reward if one of the termination conditions is satisfied. (Step e). In the other case, return to Step 2b and repeat Step 2b to Step 2d if none of the termination conditions is satisfied.

RL is conducted using the proximal policy optimization (PPO) algorithm **[71]**. After training is completed, the hit-like materials are inversely designed using the trained policy (**Figure 10**c). To empirically confirm that fragment-based RL can be a universal solution to the various problems of materials discovery with extreme properties, four case studies have been conducted. The two case studies of them are about the discovery of materials that hit multi-objective indicators used for drug discovery. The two problems are addressed in **Section 4.3** and **Section 4.4**, respectively. One of the remaining case studies is a lead molecule discovery for the 5-HT_{1B} protein target and the other is the discovery of human immunodeficiency virus (HIV) inhibitors. These are addressed in **Section 4.5** and **Section 4.6**, respectively.



	Targets	Evaluated properties				
		Round 1	Round 2	Round 3	Round 4	Round 5
logP	-0.488	2.627	2.599	1.085	-1.010	-0.488
TPSA	220.83	0.0	54.7	122.18	194.37	211.09
QED	0.207	0.517	0.774	0.660	0.371	0.205
HBA	9	1	3	6	8	10
HBD	8	0	2	4	6	8

Figure 10 Overview of fragment-based RL. a Configuration. **b** Training process. **c** Inference process. **d** BRICS combination rules.

4.4 Case Study 1: Materials Discovery with Seven-Drug Indicators

4.4.1 Problem description

Case study 1 deals with the problem of discovering materials that hit multiple target values for seven-auxiliary indicators used in drug development. The seven indicators covered here are logP, TPSA, QED, the number of hydrogen bond acceptors (HBA), the number of hydrogen bond donors (HBD), MW, and drug activity to human dopamine receptor D2 (DRD2). The drug indicators used as the target are set to be the same as the target used in the cRNN. Since cRNN is distributed as unmodifiable and training completed, fragment-based RL also used the same indicators as the target to compare the results with cRNN. The used target values are set to deviate from the distribution of common properties. Here, the common properties mean that are covered from the distribution of ChEMBL **[72, 73]** training data consisting of 1,347,173 molecular samples.

4.4.2 Configuration Setup for Fragment-based RL

Before training the policy, it is necessary to set the basic configuration of fragment-based RL: reward function, termination conditions, and combinatorial library. Here, the reward is set to be evaluated by using the error ε. ε is calculated as follows:

$$\varepsilon = \sum_{i \in set\ I} \left(\frac{y_{eval.}^{(i)} - y_{trg.}^{(i)}}{\sigma_i} \right)^2 \quad (12)$$

where $set\ I$, $y_{eval}^{(i)}$, $y_{trg}^{(i)}$, and σ_i denote a set of target indicators, evaluated property i of the molecule, target property of i, and standard deviation of i for training data, respectively. Except for DRD2, the rest of the indicators are evaluated using RDKit. In the case of DRD2, it is evaluated by a predictive model (quantitative structure-activity relationship, QSAR) provided by **ref. [33]**. From calculated ε, the reward r is evaluated as follows:

```
Potential_output = Add_Hs('Current_mol.')
if num_of_connectable_site('Current_mol.') == 0:
    if MW('Potential_output') < MW_min
        or num_of_frag. ('Potential_output') < min_frag.:
        return r = -50
    elif ε < ε_bound:
        return r = 100 / (ε+1)
    else:
        return r = 30 / (ε+1)

else:
    if MW('Potential output') < MW_max
        and num_of_frag. ('Potential_output') < max_frag.:
        if MW('Potential output') < MW_min
            or num_of_frag. ('Potential_output') < min_frag.:
        return find_next_frag. ('Current_mol.')

        elif ε < ε_bound:
            return r = 100 / (ε+1)
        else:
            return find_next_frag. ('Current_mol.')
    else:
        elif ε < ε_bound:
            return r = 100 / (ε+1)
        else:
            return r = 30 / (ε+1)
```

where ‘Current_mol.’ and ‘Potential_output’ denote the molecule in the current state and the intact molecule obtained by filling the remaining unoccupied bonding site of ‘Current_mol.’, respectively. MWmin=500 and min_frag.=6 denote a minimum molecular weight and a minimum number of fragments, respectively. These are adopted to avoid generating too small molecules. MWmax=3500 and max_frag.=50 denote the maximum molecular weight and the maximum number of fragments, respectively. These are set to terminate the game if the fragment-based RL couldn’t discover a hit-like molecule in limited rounds. ε_{bound}=0.05 denote the desired error bound. Next, it is necessary to construct a combinatorial library corresponding to the action space of the agent. The combinatorial library was constructed by selecting 2,102 BRICS molecular fragments among fragments present in BRICS 4K, which appeared more than 100 times for molecules present in the MOSES training data.

4.4.3 Experimental Setup

Before confirming whether fragment-based RL is suitable for discovering materials with extreme properties, experiments are conducted to empirically confirm that probability-distribution learning models are not suitable for discovering materials with extreme properties out of trained data distribution. probability-distribution learning models used in the experiments are two types of neural networks. One is NMT-based cRNN and the other is cVAE-based generator GCT. The data used to train both models are borrowed from **ref. [33]**, which are curated data from ChEMBL. The curated data were constructed by filtering molecules that included only H, C, N, F, S, Cl, and Br. The molecules in the curated data are drug-

like compounds that have fewer than 50 heavy atoms. The curated data are split into training data and test data. The training data includes 1,347,173 molecular samples, and the test data includes 149,679 molecular samples. **[33]**

For the trained cRNN and GCT, the general performances of the two models have been evaluated. Here, the general performance means that the target hitting ability for goal-directed inverse molecular design with target properties sampled from test data. For this, all molecular properties of the 149,679 test molecules are given as the target for the 149,679 molecular generation trials. Note that the distribution of the test set is similar to the distribution of the training set. Thus, this is closed to an evaluation of general performance for the trained area. The results of the evaluation were summarized in **Table 4**.

Root mean squared error (RMSE) can be used to evaluate how well the generated molecule satisfies each target indicator on average, however, it does not inform whether all given targets are simultaneously hit. Hence, to evaluate whether the generated molecule hits the given target properties at the same time, a new criterion is required to evaluate whether each target property is satisfied or not. In this experiment, by providing a target bound for each indicator, it is evaluated whether the property of the generated molecule is hit or not. In the phase of designing the target bounds, two methods were concerned. The first one is using a method of setting a certain percentage margin for each given target value; for example, to set each target's ±10% bound. This method can provide uniform target bounds, however, it has some drawbacks. The deviation of the bound size is quite large depending on the size of the input target value. For the evaluation of materials discovery with extreme properties, excessively large or small target bounds may be applied if a

highly extrapolated target is given. Another problem is that it does not consider the difficulty level to hit each target; the difficulty level to hit is not the same for the target indicators. Hence, the second method is adopted for the evaluation. The second method is to give ± RMSE bounds as the target bounds. It is shown in **Table 4**. The RMSE-based target bond is considering not only the difficulty level of hitting each indicator target but also the scale for each indicator. Note that the average of RMSE for each indicator shown in **Table 4** is used for the target bound for each indicator.

Figure 11 shows the training data and 10 sets of target properties (C1 to C10) out of training data distribution. If the 10 sets of target properties are set by an arbitrary combination of values for seven indicators, the sets of the target may be physically impossible. For this reason, 10 molecules were selected in the set of PubChem SARS-CoV-2 clinical trials **[74]** and their properties were adopted as target properties. Note that the set of PubChem SARS-CoV-2 clinical trials has a wider distribution than the training data. Since the 10 target sets are the sets of real molecular properties, generating molecules that hit the target sets is possible if the molecules can be derived from the combination of predefined molecular fragments. Note that the reference molecules for the 10 target sets were able to be fragmentized perfectly with the molecular fragments constituting the combinatorial library. The targets C1 to C5 are set to deviate from the logP-TPSA distribution and the targets C6 to C10 are set to deviate from the TPSA-QED distribution.

Table 4 RMSE for the target indicators and the target bounds

	logP	TPSA	QED	HBA	HBD	MW	DRD2
cRNN	0.379	5.476	0.081	0.932	0.223	5.954	0.113
GCT	0.368	5.109	0.075	1.204	0.247	8.272	0.098
Average	0.373	5.292	0.078	1.068	0.235	7.113	0.105
Target bound	±0.373	±5.292	±0.078	±1.068	±0.235	±7.113	±0.105

Table 5 Target sets (C1 to C10) for materials discovery with extreme properties

	logP	TPSA	QED	HBA	HBD	MW	DRD2
C1	13.61	293.63	0.0128	15	7	1312.84	0.0150
C2	3.31	464.92	0.0610	19	9	1269.63	0.0422
C3	-12.15	483.41	0.0682	29	18	1026.38	0.0006
C4	-7.83	810.5	0.0153	28	27	1921.81	0.2455
C5	-14.62	1447.9	0.0110	48	51	3324.74	0.0679
C6	-1.77	526.91	0.0391	20	16	1421.75	0.2228
C7	6.48	775.42	0.0289	27	27	2467.83	0.0005
C8	3.48	926.85	0.0192	31	40	2086.96	0.0315
C9	11.22	1336.88	0.0072	40	51	3123.68	0.1637
C10	-18.09	1456.35	0.0251	49	49	3106.50	0.0099

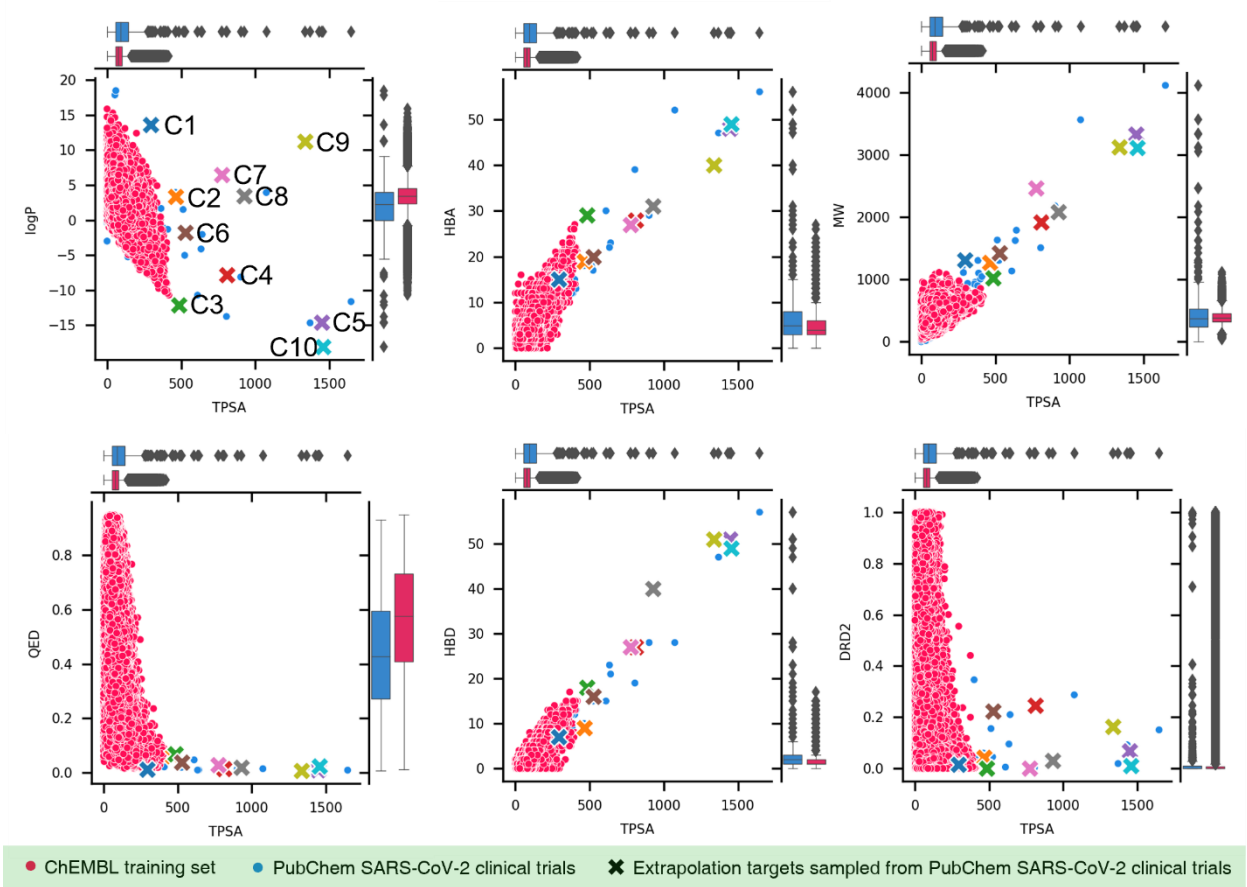


Figure 11 Training data and target sets (C1 to C10).

4.4.4 Results and Discussion

Table 6 and **Figure 12** summarize the results of trying to generate molecules for the sets of target properties (C1 to C10) that are out of the distribution of the training dataset. The results for target C0 shown in **Table 6** and **Figure 12** are the evaluated results for the 149,679 sets of target properties which are gathered from molecules in the test set.

Molecular generations have been attempted 10,000 times for each set of target properties (C1 to C10). **Table 6**a shows how many valid molecules were generated among 10,000 generation attempts. Here, the valid molecules denote the molecules that do not violate the chemical valence rule. The percentage number in parentheses means the percentage of unique molecules among the generated valid molecules. **Table 6**b shows the number of valid molecules that simultaneously satisfy the target bounds of logP, TPSA, QED, HBA, and HBD, excluding MW and DRD2 targets among 10,000 generation attempts. In parentheses, the percentage of unique molecules among them is written down. **Table 6**c shows the number of generated molecules that simultaneously satisfied all seven targets, including MW and DRD2 targets, among 10,000 molecule generation attempts. The ratio of unique molecules among them is written in parentheses. **Table 6**d-j shows the number of molecules that have met the target bound of each single target value among 10,000 molecule generation attempts.

The results of target set C0 are the evaluated results for the molecular generation with 149,479 target sets gathered from 149,479 reference molecules in the test set.

The results for target set C0 were rescaled to compare; 149,479 → 10,000. These show the general performance of the inverse molecular design with cRNN. As can be seen in **Table 6** and **Figure 12**, both probability distribution-learning models showed good performance for the inverse molecular design that hit the given target properties C0. cRNN generated valid molecules in 84.8% of the 149,479 molecular attempts. In addition, 34.8% of trials succeeded to generate molecules that hit all seven target properties at the same time, and 44.5% of the total trials generated molecules hitting five target properties except for MW and DRD2 at the same time. In the case of GCT, valid molecules were generated in 87.2% of the 149,479 generation attempts. In addition, 26.64% of the total attempts succeeded to generate molecules hitting the given seven target properties simultaneously. 39.9% of total generation trials succeeded to hit the five targets except for MW and DRD2. It means that the probability-distribution learning models work well for the targets distributed in their trained area. However, the models failed to generate molecules for the targets that deviated from the trained data distribution (see results C1 to C10 in **Table 6** and **Figure 12**), which corresponds to the materials discovery with extreme properties. In contrast, it is confirmed that fragment-based RL is possible to some extent for inverse molecular design with extreme targets. For C1, C2, C3, C4, and C6, molecules that simultaneously satisfy all the seven target properties were discovered. In the cases of generation for targets C5, C7, C8, C9, and C10, any molecule that hits all the seven target properties at the same time was not discovered, but the molecules that hit the five target properties (logP, TPSA, QED, HBA, and HBD) simultaneously were discovered. Therefore, it is believed that fragment-based RL can solve the materials discovery problem with extreme properties to some extent, which is not possible with probability-distribution learning models.

Then, the question remains why the fragment-based RL failed to generate for the targets C5 and C7 to C10. As can be seen in **Figure 11**, the target C5 and C7 to C10 deviate more from the distribution of the trained data than the other targets (succeeded targets: C1 to C4 and C6). The farther the distance from the center where the CheMBL data are distributed, the farther it is from the properties of materials commonly found in nature. For example, the target value of logP for target set C10 is -18. It means that materials to be discovered have an affinity to water 10^{18} times more than lipids. This is not a common characteristic of a fairly large molecule with a molecular weight of 3106.5 g/mol (target MW for target set C10). The target C5 is similar; -14.6 logP for 3324.7 g/mol MW. That is, considering that the unsuccessful targets are not commonly found in nature, it is believed that the probability that molecules—generated by the combination of randomly selected molecular fragments—have such target properties is low. Considering that fragment-based RL is a model that starts learning from random molecular fragment selection and combination at the beginning of learning, the frequency of obtaining a good reward may be low.

Figure 13 shows the change in the maximum reward obtained at each training step. For the succeeded targets C1 to C4 and C6, it is confirmed that a high maximum reward was obtained frequently during the training. However, a relatively low maximum reward and low frequency of hitting high reward were observed for the unsuccessful targets C5 and C7 to C10. Interestingly, for targets C7 and C8, there was no significant difference in the maximum reward compared to those of successful targets. However, the frequency of obtaining the maximum reward near 100 was relatively low compared to those of the successful targets. When an

additional 10,000 attempts were made for each unsuccessful target, one molecule was found that simultaneously hits all the target properties for C8. On the other hand, for targets C5, C9, and C10, a significantly lower maximum reward was observed compared to those of the successful targets. When looking at the above-mentioned results comprehensively, the number of occurrences of events that can obtain a desired level of reward is relatively poor for the more extreme targets. Such a problem is called a sparse reward problem, the frequency of obtaining the desired level of reward from the environment is low. Studies to solve or alleviate the sparse reward problem are steadily progressing. Thus, this study left the issue of the sparse reward problem as a point for improvement in future studies.

Additionally, it is possible to find the reason from the point of view of the target bound. The target bounds of MW and DRD2 used as evaluation criteria are ±7.113 and ±0.105, respectively. The target bound of MW is too tight to include even a single atom error. Considering the target MWs of failed targets (C5 and C7 to C10) were in the range of 2,000 to 3,000 g/mol, the target bound of MW was set to tight. If the target bound is increased, the number of molecules—generated by fragment-based RL—that are evaluated as hitting the target bounds are increased. This can be confirmed through the design error (pRMSE) for each target property of the generated molecules shown in **Figure 12**. That is, the criterion for determining the success or failure of molecular discovery is highly dependent on the size of the target bound. Thus, it is difficult to conclude that the molecular generations for the targets C5 and C7 to C10 failed. However, be aware that it is difficult to conclude that the failure reason for cRNN and GCT is also related to the issue of target bound size. Because all the molecules they generated were molecules such as methane or ethane

with one or two heavy atoms. Just considering the level of target MW, it didn't seem to be working correctly.

There are also concerns regarding the accuracy of the QSAR model used to evaluate DRD2. In general, the QSAR model that evaluates drug activity from a molecular structure is not very accurate. In addition, even in the cRNN paper, target hitting ability for DRD2 was not evaluated. In the paper, the QSAR model for DRD2 was used as a binary classifier. The test molecules are evaluated as active if the QSAR model of DRD2 predicted a value of 0.5 or higher. The others are evaluated as inactive. Therefore, using MW and DRD2 as evaluation criteria may cause problems in the objective analysis of model performance. For this reason, in Case Study 2, the inverse molecular design was performed for the five target properties except for MW and DRD2.

Table 6 Molecular generation results for extreme targets and interpolation targets

Model	Target type	Target	a. # of valid mol. (% of unique mol.)	b. # of all prop. hit w/o MW& DRD2 (% of unique mol.)	c. # of all prop. hit (% of unique mol.)	d. logP	e. TPSA	f. QED	g. HBA	h. HBD	i. MW	j. DRD2
RL (ChEMBL)	Extreme targets	C10	10,000 (100%)	23 (100%)	0 (- %)	820	1,133	9,978	5,533	2,618	0	35
		C9	10,000 (100%)	289 (83%)	0 (- %)	3,919	1,432	9,995	7,880	3,223	1	15
		C8	10,000 (100%)	181 (100%)	0 (- %)	2,999	974	9,992	6,273	2,379	466	133
		C7	10,000 (100%)	321 (100%)	0 (- %)	3,686	1,420	9,971	7,832	3,588	531	28
		C6	10,000 (100%)	1,708 (100%)	311 (100%)	3,769	3,450	9,890	8,636	7,301	1,445	8,729
		C5	10,000 (100%)	14 (100%)	0 (- %)	1,607	1,377	9,981	7,381	2,363	0	26
		C4	10,000 (100%)	439 (100%)	50 (100%)	1,972	1,452	9,809	7,283	4,384	572	9,493
		C3	10,000 (99%)	913 (96%)	233 (96%)	3,143	2,296	9,940	8,392	7,143	1,128	8,310
		C2	10,000 (100%)	2,424 (100%)	366 (100%)	6,738	3,145	9,928	8,415	8,635	1,388	4,970
		C1	10,000 (100%)	1,317 (100%)	355 (100%)	5,202	2,282	9,918	9,347	8,622	1,849	9,406
cRNN (ChEMBL)	Extreme targets	C10	422 (- %)	0 (- %)	0 (- %)	422	0	0	0	0	0	422
		C9	0 (- %)	0 (- %)	0 (- %)	0	0	0	0	0	0	0
		C8	0 (- %)	0 (- %)	0 (- %)	0	0	0	0	0	0	0
		C7	0 (- %)	0 (- %)	0 (- %)	0	0	0	0	0	0	0
		C6	6 (100%)	0 (- %)	0 (- %)	0	0	6	1	0	0	5
		C5	0 (- %)	0 (- %)	0 (- %)	0	0	0	0	0	0	0
		C4	9,836 (- %)	0 (- %)	0 (- %)	0	0	0	0	0	0	0
		C3	3,022 (- %)	0 (- %)	0 (- %)	0	0	4	2	1	0	3,022
		C2	0 (- %)	0 (- %)	0 (- %)	0	0	0	0	0	0	0
		C1	1,768 (- %)	0 (- %)	0 (- %)	0	0	4	1	1	0	1,788
		*C0	*8,475 (100%)	*4,453 (100%)	*3,479 (100%)	*7,391	*7,732	*7,622	*9,001	*9,531	*8,139	*9,200
GCT (ChEMBL)	Extreme targets	C10	0 (- %)	0 (- %)	0 (- %)	0	0	0	0	0	0	0
		C9	0 (- %)	0 (- %)	0 (- %)	0	0	0	0	0	0	0
		C8	0 (- %)	0 (- %)	0 (- %)	0	0	0	0	0	0	0
		C7	0 (- %)	0 (- %)	0 (- %)	0	0	0	0	0	0	0
		C6	0 (- %)	0 (- %)	0 (- %)	0	0	0	0	0	0	0
		C5	0 (- %)	0 (- %)	0 (- %)	0	0	0	0	0	0	0
		C4	0 (- %)	0 (- %)	0 (- %)	0	0	0	0	0	0	0
		C3	3 (100%)	0 (- %)	0 (- %)	0	0	3	1	0	0	3
		C2	0 (- %)	0 (- %)	0 (- %)	0	0	0	0	0	0	0
		C1	4 (100%)	0 (- %)	0 (- %)	0	0	1	0	0	0	4
		*C0	*8,715 (99%)	*3,994 (99%)	*2,664 (100%)	*7,217	*7,937	*7,680	*8,048	*9,409	*6,759	*9,312

*Experimental results for 149,679 targets obtained from molecules in the test set. The values are rescaled: 149,679 → 10,000

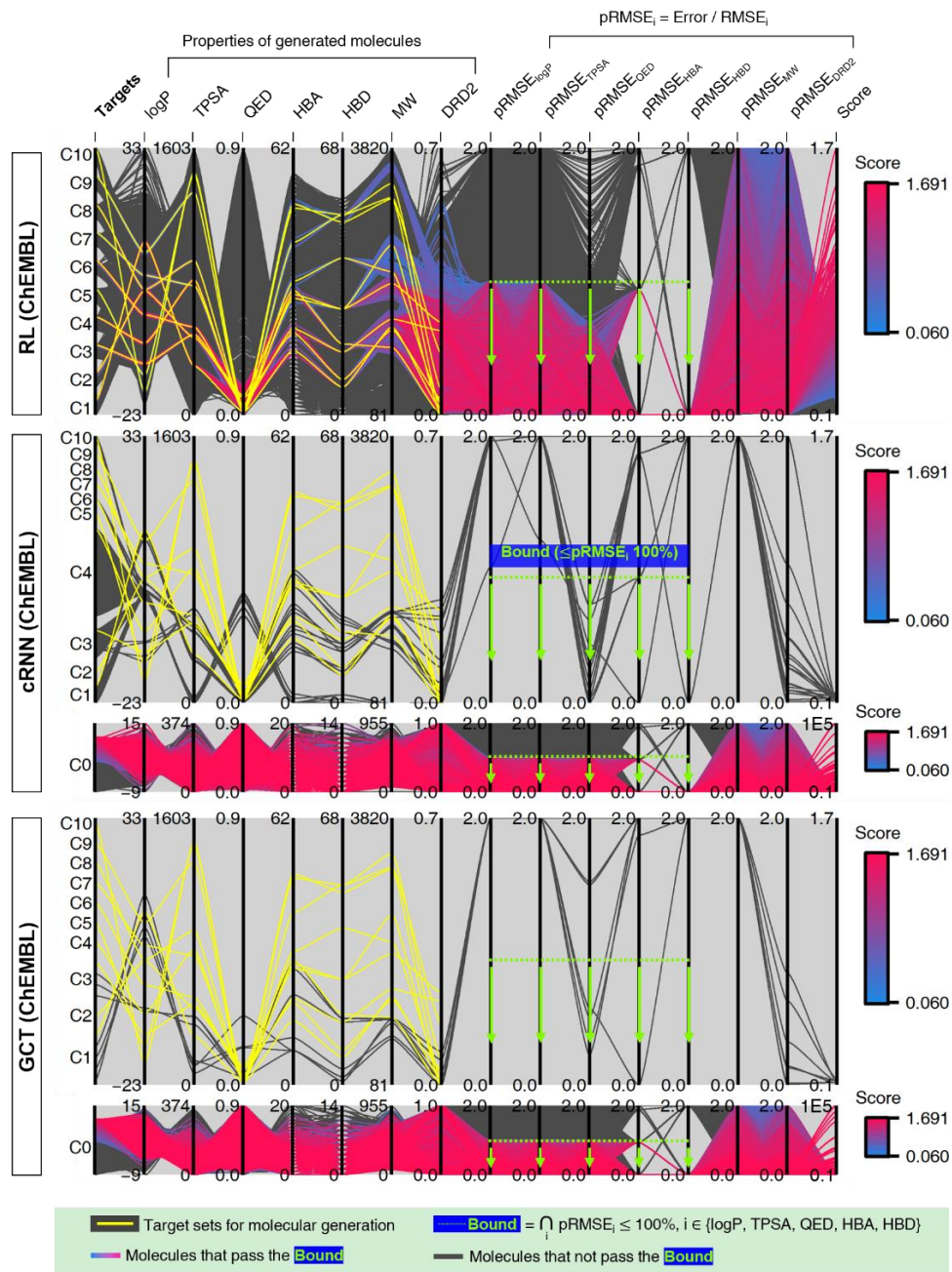


Figure 12 Parallel coordinates plots for comparison of materials discovery performance between fragment-based RL and probability-distribution learning models. Yellow lines indicate target properties. Each blue-red colored line indicates a set of properties of a molecule that hits the *Bound* (green). Here, *Bound* indicates the intersection of five target bounds of logP, TPSA, QED, HBA, and HBD. Each grey-colored line indicates a set of properties of a molecule that does not hit the *Bound*. Score indicates $\sum_i \mathbf{1}/\boldsymbol{pRMSE_i}$.

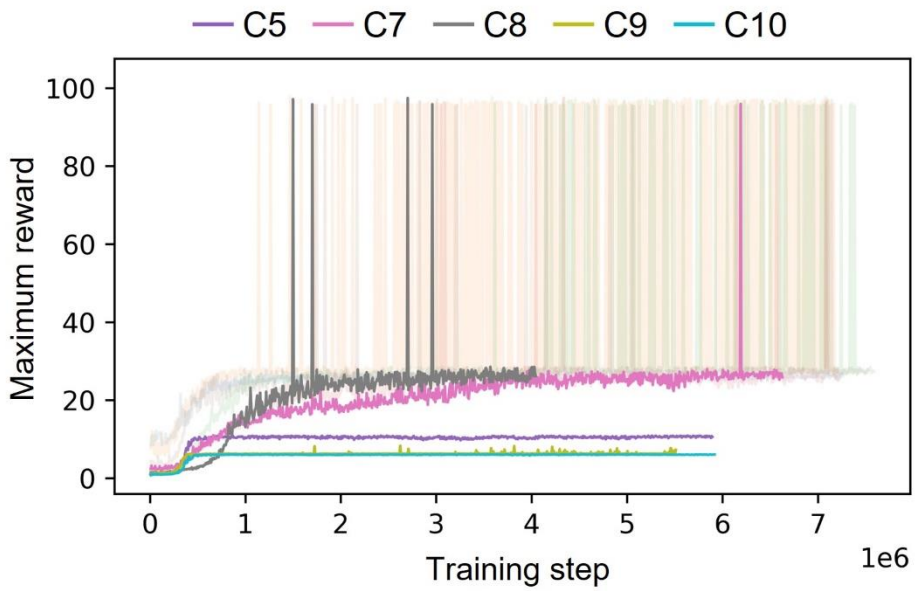


Figure 13 Maximum reward plots for training steps. Opaque lines denote the maximum reward curve for failed targets (C5, C7, C8, C9, and C10) and the transparent lines denote the maximum reward curve for successful targets C1, C2, C3, C4, and C6.

4.5 Case Study 2: Materials Discovery with Five-Drug Indicators

4.5.1 Problem description

Case study 2 is a supplementary experiment to Case study 1. Case study 2 is conducted for the target logP, TPSA, QED, HBA, and HBD. Here, MW and DRD2 which caused problems in model evaluation in Case study 1 are not considered. Since the training code of cRNN is not disclosed and released as a trained state, so modifying the network structure and re-training is impossible. Hence, Case study 2 is conducted only on GCT and fragment-based RL. Case study 2 uses the MOSES data set.

4.5.2 Configuration Setup for Fragment-based RL

In this section, error ε and reward r are calculated in the same way as in the previous section. The parameter values applied here are MWmax=500, MWmin=50, max_frag.=10, min_frag.=2, and ε_{bound}=0.1. The combinatorial library consists of 2,102 molecular fragments among BRICS 4K fragments. The sampled molecular fragments appear more than 150 times for molecules present in the MOSES training set.

4.5.3 Experimental setup

The MOESS training data and target sets (M1 to M10) to be used in this section are shown in **Figure 14**. 10 Target sets are the sets of properties gathered from 10 molecules in a curated data set. The data set is curated from PubChem anti-cancer molecules with a molecular weight of 500 g/mol or less. The exact target values for the 10 target sets are shown in **Table 7**. The general performance for M0—176,074 targets gathered from 176,074 molecules in the MOSES test set—is also evaluated. The RMSE for each indicator is summarized in **Table 8**. Since the min-max scale of each indicator in the MOSES training set is changed, the target bounds are reset as the RMSE summarized in **Table 8**.

Table 7 Target sets (M1 to M10) for materials discovery with extreme properties

	logP	TPSA	QED	HBA	HBD
M1	7.28884	47.67	0.281783	4	1
M2	6.45712	90.54	0.274407	5	2
M3	5.18752	123.57	0.221414	9	1
M4	3.8689	155.3	0.255532	5	5
M5	1.947	186.58	0.421275	10	2
M6	1.0672	50.41	0.192539	2	3
M7	4.6269	97.15	0.17826	8	1
M8	5.71644	139.78	0.184438	5	4
M9	0.71837	178.24	0.212291	9	4
M10	-0.48843	220.83	0.206826	9	8

Table 8 RMSE for the target indicators and the target bounds

	logP	TPSA	QED	HBA	HBD
GCT	0.214	3.225	0.037	0.180	0.106
Target bound	±0.214	±3.225	±0.037	±0.180	±0.106

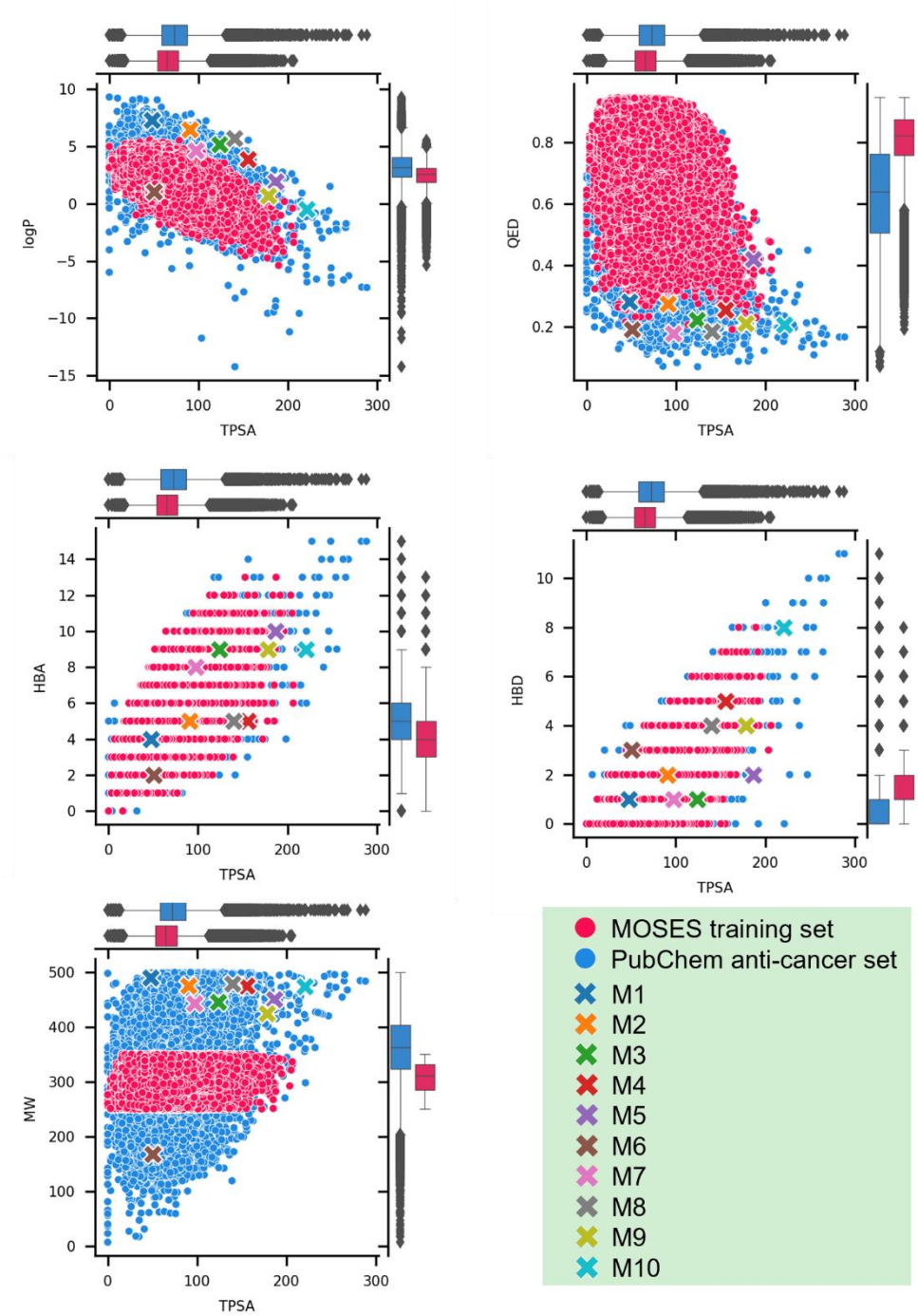


Figure 14 MOSES training data and target sets (M1 to M10).

4.5.4 Results and Discussion

The experimental results are summarized in **Table 9** and **Figure 15**. In this experiment, it was confirmed that the general performance of GCT is good (for target M0). In the case of GCT, valid molecules were generated in 97.5% of the 176,074 generation attempts. In addition, 53.7% of the total attempts succeeded to generate molecules hitting all five target properties simultaneously. It means that the probability-distribution learning models work well for the targets distributed in their trained area. However, GCT almost failed to generate hit-like materials with target properties out of trained data distribution, except for target M9. For target M9, six attempts succeeded out of 10,000 generation attempts. However, there were only two unique molecules. In contrast, fragment-based RL succeeded to generate molecules that hit all five target properties simultaneously for all the target sets except M6. The reason why fragment-based RL failed in molecular design for the target set M6 lies in the composition of the combinatorial library. **Figure 15**b shows the molecular weight distribution of the molecular fragments constituting the combinatorial library and the molecular weight distribution of the reference molecules for each target set (M1 to M10). The molecular weight of the reference molecule corresponding to M6 is 167 g/mol. In fact, since logP, TPSA, HBA, and HBD are highly correlated with molecular size, the molecules to be generated must have an MW of approximately 167 g/mol to hit the target properties of M6. However, the molecular weight of some molecular fragments was larger than this. In addition, since the initial fragment was randomly selected, there may be insufficient room to match the target. Therefore, it seems necessary to consider molecular weight when constructing the combinatorial library when the target material is small.

Table 9 Molecular generation results for extreme targets and interpolation targets

Model	Target type	Target	a. # of valid mol. (% of unique mol.)	b. # of all prop. hit (% of unique mol.)	d. logP	e. TPSA	f. QED	g. HBA	h. HBD
RL (MOSES)	Extreme targets	M10	10,000 (81%)	949 (67%)	4,345	3508	5,429	5,482	6,858
		M9	10,000 (92%)	1,767 (90%)	6,143	5186	4,050	7,867	7,897
		M8	10,000 (98%)	4,108 (96%)	6,159	5194	6,320	7,007	6,984
		M7	10,000 (80%)	271 (66%)	6,232	4630	721	8,140	8,847
		M6	10,000 (48%)	0 (- %)	3,158	2085	5	2,121	2,500
		M5	10,000 (77%)	46 (61%)	8,519	7455	234	9,158	9,505
		M4	10,000 (92%)	3,240 (89%)	7,176	6434	5,159	7,998	8,333
		M3	10,000 (94%)	1,325 (92%)	7,289	6618	1,972	8,094	8,883
		M2	10,000 (98%)	4,432 (97%)	5,970	5679	6,732	7,407	7,730
		M1	10,000 (86%)	3,016 (82%)	5,363	5021	6,340	7,772	8,118
GCT (MOSES)	Extreme targets	M10	2,014 (3%)	0 (- %)	43	427	1,383	570	218
		M9	4,603 (7%)	6 (33%)	1,598	878	109	1,732	3,542
		M8	1,788 (2%)	0 (- %)	263	1	163	1,418	1,378
		M7	3,743 (17%)	0 (- %)	1,695	433	34	2,392	2,747
		M6	5,645 (6%)	0 (- %)	1,147	2188	50	3,778	4,914
		M5	1,255 (2%)	0 (- %)	1,151	25	4	63	1,028
		M4	1,498 (4%)	0 (- %)	32	235	24	217	1,143
		M3	1,185 (6%)	0 (- %)	133	42	0	284	553
		M2	5,084 (3%)	0 (- %)	2,151	282	772	2,385	2,463
		M1	5,026 (2%)	0 (- %)	207	902	106	2,862	4,908
	Common targets	*M0	9,749 (96%)	5,365 (92%)	7,058	7,650	8,572	9,436	9,639

Experimental results for 149,679 targets were obtained from molecules in the test set. The values are rescaled: 176,074 → 10,000

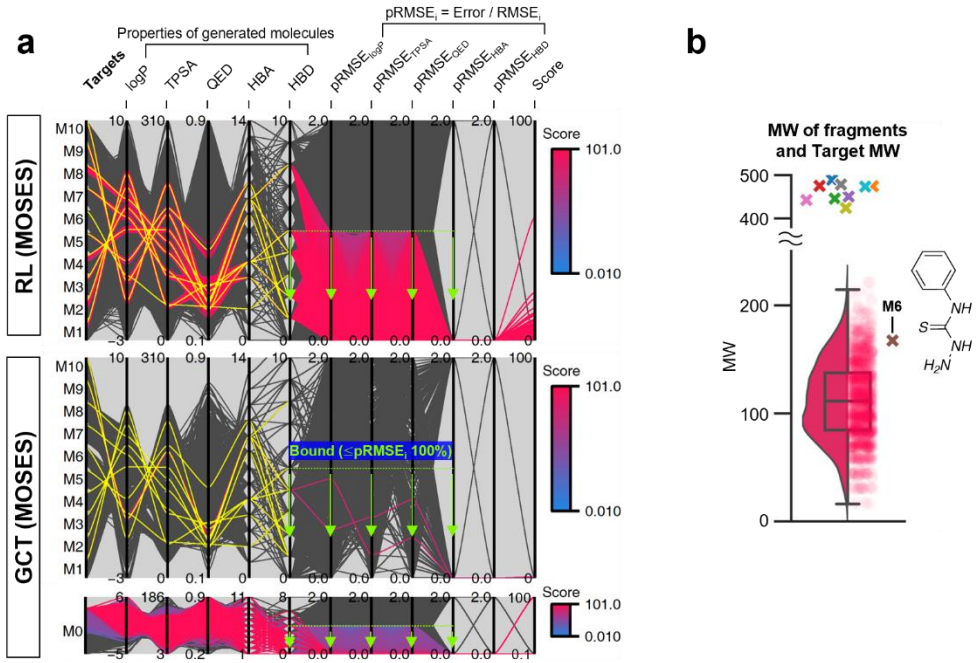


Figure 15 Experimental results for Case study 2. a Parallel coordinates plots for comparison of materials discovery performance between fragment-based RL and probability-distribution learning models**.** Yellow lines indicate target properties. Each blue-red colored line indicates a set of properties of a molecule that hits the *Bound* (green). Here, *Bound* indicates the intersection of five target bounds of logP, TPSA, QED, HBA, and HBD. Each grey-colored line indicates a set of properties of a molecule that does not hit the *Bound*. Score indicates $\sum_i 1/pRMSE_i$. **b** Distribution of molecular weights for molecular fragments and source molecules of the target set M1 to M10.

4.6 Case study 3: Ligands Discovery for Ligand—5-HT_{1B} Protein Receptor Docking

4.6.1 Problem description

Case study 3 deals with the discovery of molecules that dock to 5-hydroxytryptamine receptor 1B (5-HT_{1B}), a protein encoded by the human HTR1B gene. **[69]** 5-HT_{1B} receptors are widely distributed throughout the human central nervous system and are known to play different roles depending on their location. **[74-79]** In particular, the 5-HT1B receptor has been reported that inhibits the release of various neurotransmitters such as dopamine, serotonin, gamma-aminobutyric acid, acetylcholine, and glutamate. **[80]** It is also related to vasoconstriction, migraine treatment, and major depressive disorder treatment. **[78, 81-84]** In this chapter, the fragment-based RL is trained to discover materials that are likely to dock to the 5-HT_{1B} receptor.

4.6.2 Setup for Fragment-based RL

The task of Case Study 3 is to discover docking materials that maximize the binding affinity to the 5-HT_{1B} receptor. In this problem, Quick Vina 2 (QVina2) **[85]**—which can quickly calculate a docking score that is inversely proportional to binding affinity—is used as a docking score evaluator; The docking score is inversely proportional to the binding affinities. QVina2 searches the optimal conformation of docking material in the simulation box using an optimization algorithm and evaluates the docking score for the found optimal conformation. The

exhaustiveness parameter—which plays a role that is similar to the population of the genetic algorithm—is set to eight. The reward is evaluated as a negative docking score and RL finds the policy that maximizes the reward.

4.6.3 Results and Discussion

10,000 attempts have been made to design ligand materials using the trained fragment-based RL and 9,369 unique ligands have been obtained. **Figure 16**a shows the results of comparing the docking score between the designed ligands and reference molecules. The reference molecules were obtained by random sampling of 10,000 molecules among the ChEMBL test set used in Case study 1. Note that the ChEMBL test set consists of drug-like molecules. The median docking score of the designed ligands was -12.9, and the minimum value was -18.1. On the other hand, the median docking score of the reference molecules was -9.2, and the minimum value was -14.2; Note that the lower the calculated docking score, the better to dock to the protein receptor. It means that it could be more effective to discover the potential drugs with fragment-based RL rather than screening the known reference materials.

Furthermore, it was tried to find the matched ChEMBL materials with the designed ligands. Since the some of ChEMBL materials are labeled with experimental results of drug activity, if we could find the active ligands among the matched molecules, it means that some of the unidentified generated ligands would the potential to be used as new drugs. When checking whether there are matched molecules in ChEMBL, seven matched molecules were found. Unfortunately, there

is no drug activity labeled molecule among the seven. Since the number of drug-like molecules that a molecular weight of less than 500 Da is estimated bigger than 10^{60}, finding the matched and labeled molecules among 2.2 million molecules constituting ChEMBL would be hard to succeed. Hence, when trying to search the matched molecules with 100% of Tanimoto similarity, 18 matched molecules were found. Five among the 18 matched molecules had active labels. Note that there are countless kinds of drug targets, and the experiment results of drug activity are labeled with only the tested target. Hence, the rest of them—the remaining 13 molecules—does not mean these are inactive to the 5-HT_{1B} receptor. In fact, for the five active drugs, the labeled targets were not 5-HT_{1B}. However, three of them target the protein receptor of Homo sapiens and the target disorders were similar to the target disorders of 5-HT_{1B} inhibitors; the others target fungus. It implies that the designed ligands have the potential. The results are summarized in **Figure 16**b-f and **Table 10**.

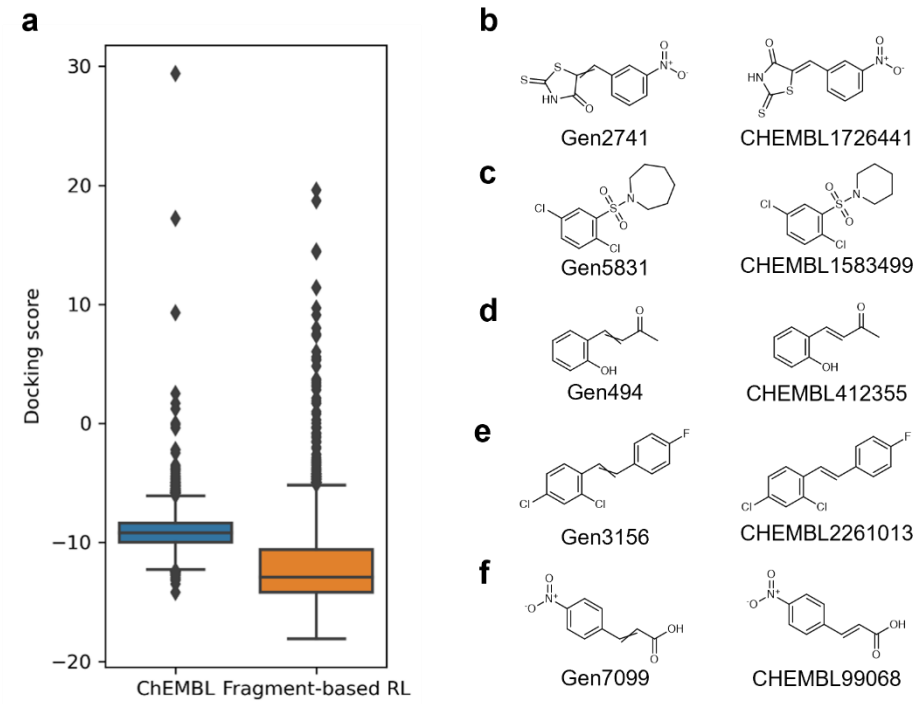


Figure 16 Results of lead molecules discovery for 5-HT$_{1B}$ receptor. a Boxplot comparison between 10,000 molecules randomly sampled from the curated ChEMBL test set and 10,000 molecules generated by fragment-based RL. **b-f** The five Chembl molecules that show 100% Tanimoto similarity with 10,000 designed molecules. The left indicates molecules that were generated by fragment-based RL and the right indicates molecules that matched ChEMBL molecules.

Table 10 Drug activity of matched molecules

Matched molecule	Drug activity	Description	Target disorder
CHEMBL1726441	Active	NRF2 inhibitor	Neurodegenerative diseases (Parkinson's disease; chronic central nervous system degenerative diseases caused by progressive loss of dopaminergic neurons)
	Active	TDP1 inhibitor	Neurodegenerative disorder with axonal neuropathy
	Active	GMMN	Involved in cell replication (anticancer drug target), self-renewal and survival of enteric nervous system progenitor cells related
	Active	EPAC2 antagonist	Traumatic brain, spinal cord, and nerve damage insulin secretion promotion, heart failure, dopamine release related
CHEMBL1583499	Active	TDP-43 Inhibitors	Accumulation of TDP-43 aggregates in the central nervous system causes neurodegenerative diseases such as amyotrophic lateral sclerosis (ALS), frontotemporal dementia (FTD), and Alzheimer's disease (AD) (a target for the treatment of neurodegenerative diseases)
CHEMBL412355	Active	LMP-1 inhibitors	Latent membrane protein-1 (LMP-1) encoded by Epstein-Barr virus (EBV) induces unregulated cell growth (EBV-related anticancer drug target)
CHEMB2261013	Active	Antifungal activity against Penicillium chrysogenum	Not human target
CHEMBL99068	Active	Sporosarcina pasteurii CCM 2056 urease inhibitor	Not human target

5-HT_{1B} receptor:

- Distributed throughout the central nervous system **[86-88]**
- Control the release of neurotransmitters such as acetylcholine, glutamate, dopamine, noradrenaline, and gamma-aminobutyric acid **[89]**
- Drug target for major depressive disorders **[81,82]**
- Facilitation in excitatory synaptic transmission **[90]**
- Pulmonary vasoconstriction to treat migraines **[83,84]**
- Increase the osteoblasts, bone mass, and bone formation rate **[91-95]**

4.7 Conclusion

Chapter 3 describes the problem of materials discovery hitting extreme and rare properties out of known materials' distribution. Here, it has been theoretically and empirically demonstrated that inverse molecular design models based on probability-distribution learning (e.g., NMTs, VAEs, and GANs) are improper to design materials with extreme properties out-of-training-data-distribution. In contrast, fragment-based RL can design materials with extreme properties, which cannot be solved using probability-distribution learning models. It has been demonstrated that the fragment-based RL can solve various materials discovery problems to find better materials: multi-objective materials discovery and specific property maximized materials discovery for ligand-protein docking. Considering these results, it is believed that the fragment-based RL could be a universal solution to the problem of materials discovery with extreme properties out of known data distribution. In addition, the limitations of fragment-based RL regarding an issue of constructing the combinatorial library and another issue regarding the sparse reward problem have been discussed.

Chapter 5. Conclusion

5.1 Summary of contributions

This thesis has studied deep learning-aided materials discovery that can be used to accelerate the process of materials development. Two kinds of goal-directed inverse molecular design models have been proposed. One is the generative chemical Transformer (GCT), a generative model that embeds a Transformer in cVAE (addressed in Chapter 3). It can generate realistic and chemically feasible molecules hitting a set of target properties by understanding the sequential meaning of chemical language. This study has shown the advantages of adopting the attention mechanism-based natural language model to the generative model. It is confirmed that the attention mechanism helps to understand the molecular structure hidden in the line notated chemical language by attention to the tokens constituting the chemical string. High attention scores have been observed for the structurally related key tokens from given query tokens. Considering the results, it has been confirmed that the attention mechanism in the natural language model helps to understand the molecular structural meaning hidden in one-dimensional chemical language. Furthermore, when considering the benchmarking results that the validity of generated molecules with GCT—an attention mechanism-based model—was higher than those of the other language-based models that do not use an attention mechanism, it is believed that the structural understanding based on attention mechanism helps the chemically feasible molecular generation. In addition, when considering the results that the substructural patterns of the generated molecules and real molecules were shared and it makes the generated molecules seem more realistic, the ability to understand the

patterns of corpus obtained from the natural language model helps the molecular generation be more realistic. Note that it has been quantitatively analyzed that the similarity between the generated molecules and the real molecules is quite high. In conclusion, the study of GCT—an attention mechanism-based language model embedded generative model—has shown the advantages of embedding a natural language model in the molecular generator, realistic and chemically feasible molecular generation.

The other goal-directed inverse molecular generation model (addressed in Chapter 4) is a fragment-based RL that can generate the molecules with a set of extreme and rare target properties, which are not possible with probability-distribution learning such as NMTs, VAEs, and GANs-based inverse molecular design models. This study has theoretically and empirically shown that the probability-distribution learning models are not proper to the molecular generation out-of-training-data-distribution. It regards the problems of materials discovery with extreme properties that were not observed or rarely observed. Since the probability-distribution learning models are trained to obtain a generator that approximates the empirical probability of training data, the models are not proper to generate data out of training data distribution. The true probability of the considering system cannot be known and the empirical probability of training data cannot be equal to the true probability. That is because the training data are gathered only for the observable cases. Hence, the probability-distribution learning models are not proper to generate the molecules out of training data distribution. In contrast, it is confirmed that the proposed fragment-based RL—a goal-directed molecular generator that generates molecules by combining molecular fragments—can design molecules with a set of extreme and rare properties. That is because combinatorial chemistry—a base

environment of the fragment-based RL—can generate all possible molecules that can be obtained by random combinations of predefined molecular fragments. Furthermore, this study has solved the disadvantage of combinatorial chemistry by giving an AI-driven policy that guides the selection of molecular fragment which is proper to combine to get the desired products. In addition, it has been demonstrated that the fragment-based RL is applicable to various materials discovery problems from multi-objective molecular design to molecular design with maximizing certain properties. It is believed that the ability of molecular generation out of observed samples' distribution will contribute to the practical materials discovery problems to find better materials than already discovered materials.

In conclusion, considering that the goal-directed inverse molecular design models do their work in sub-seconds, it is believed that the deep-learning aided materials discovery will contribute to accelerating the materials development process.

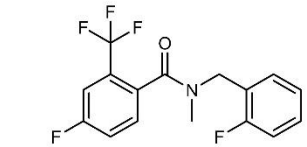
6.2 Challenges and opportunities

In the field of process system engineering, there have been a lot of efforts to integrate the system from the product design to the process design. Since the 1980s, especially AI (e.g., knowledge-based systems and expert systems) adopted studies to integrate multi-domain knowledge and drive the integrated system autonomously have been reported. **[96-100]** Although many meaningful results have been reported, it is still hard to design intact hand-craft knowledge. Even for a problem that human answers easily (e.g., classification of cat and dog images), it is hard to be coded.

However, in the recent decade, deep learning and machine learning have shown that data-driven models can solve problems by recognizing the patterns of data without hand-crafted features. Due to this big advance of data-driven AI, many problems linking the traditional models that were hard to be coded are solved, e.g., goal-directed inverse molecular design, product characterization, recommendation of operating conditions, experimental condition planning, theory discovery, and self-decision making. **[101-106]** It enables the integration of processes from the product design to the process design. In more recent years, successful research on AI-driven integrated system have begun to be reported, e.g., self-driving laboratory and AI-driven flow chemistry systems. **[107-109]** Therefore, it is believed that AI-aided materials discovery will greatly contribute to material science research.

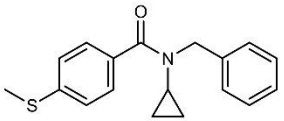
Appendix: Results of molecular generation using GCT

The four numbers below the molecular image indicate logP, TPSA, QED, and the number of tokens in SMILES, respectively.



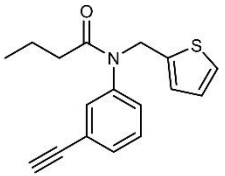
CN(Cc1ccccc1F)C(=O)c1ccc(F)cc1C(F)(F)F

4.2558, 20.31, 0.7709, 38



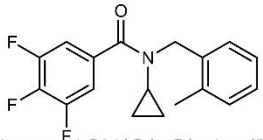
CSc1ccc(C(=O)N(Cc2ccccc2)C2CC2)cc1

4.2133, 20.31, 0.7704, 34



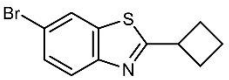
C#Cc1cccc(N(Cc2cccs2)C(=O)CCC)c1)CC1

4.0627, 20.31, 0.7577, 32



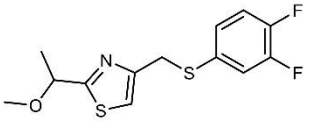
Cc1ccccc1CN(C(=O)c1cc(F)c(F)c(F)c1)C1CC1

4.2171, 20.31, 0.7729, 40



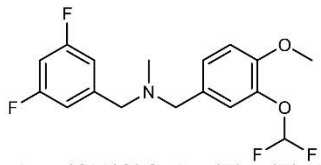
Brc1ccc2nc(C3CCC3)sc2c1

4.3263 12.89 0.7499, 22



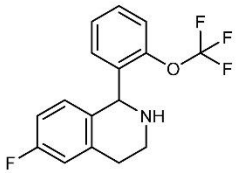
COC(C)c1nc(CSc2ccc(F)c(F)c2)cs1

4.421, 22.12, 0.7632, 31



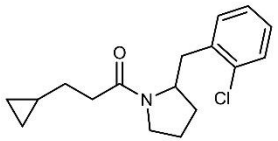
COc1ccc(CN(C)Cc2cc(F)cc(F)c2)cc1OC(F)F

4.2068, 21.70, 0.7033, 38



Fc1ccc2c(c1)CCNC2c1ccccc1OC(F)(F)F

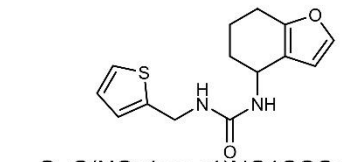
3.9594, 21.26, 0.8478, 34



O=C(CCC1CC1)N1CCCC1Cc1ccccc1Cl

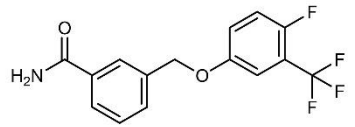
4.0637, 20.31, 0.8009, 29

Figure S1 Generated molecules with target condition #1 shown in Figure 7.



O=C(NCc1cccs1)NC1CCCc2o
ccc21

3.2179, 54.27, 0.9040, 28

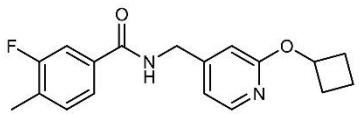


NC(=O)c1cccc(COc2ccc(F)c(C(F)(F)F)
c2)c1

3.5224 52.32 0.8781, 39

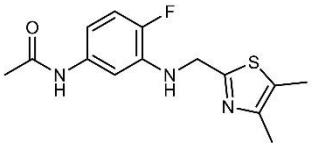
CC1CCN(C(=O)NCc2ccc(OCC(F)(F)F
)nc2)CC1C

3.2102, 54.46, 0.9116, 39



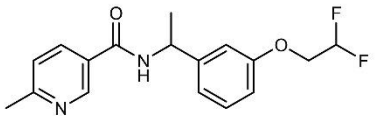
Cc1ccc(C(=O)NCc2ccnc(OC3
CCC3)c2)cc1F

3.3904, 51.22, 0.9201, 36



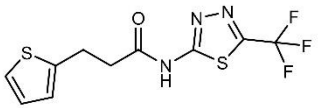
CC(=O)Nc1ccc(F)c(NCc2nc(C)c(C)s
2)c1

3.46954 54.02 0.907246, 35



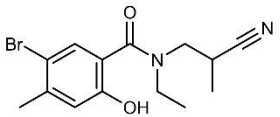
Cc1ccc(C(=O)NC(C)c2cccc(OCC(F)
F)c2)cn1

3.5249, 51.22, 0.8857, 38



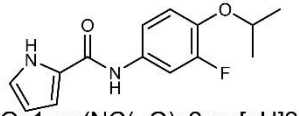
O=C(CCc1cccs1)Nc1nnc(C(F)(F)F)s
1

3.1897, 54.88, 0.9435, 32



CCN(CC(C)C#N)C(=O)c1cc(Br)c(C)
cc1O

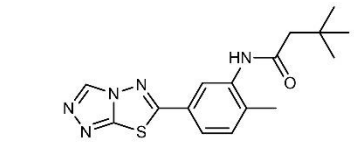
3.0849, 64.33, 0.9251, 33



CC(C)Oc1ccc(NC(=O)c2ccc[nH]2)cc
1F

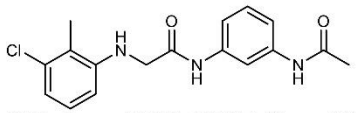
3.1933, 54.12 0.8886, 30

Figure S2 Generated molecules with target condition #2 shown in Figure 7.



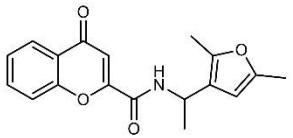
Cc1ccc(-c2nn3cnnc3s2)cc1NC(=O)CC(C)(C)C

3.5359, 72.18, 0.7976, 39



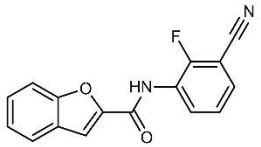
CC(=O)Nc1cccc(NC(=O)CNc2cccc(Cl)c2C)c1

3.6574, 70.23, 0.7827, 37



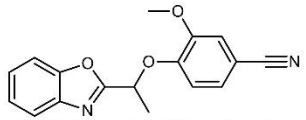
Cc1cc(C(C)NC(=O)c2cc(=O)c3ccccc3o2)c(C)o1

3.4938, 72.45, 0.8033, 41



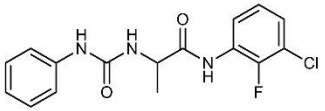
N#Cc1cccc(NC(=O)c2cc3ccccc3o2)c1F

3.6959, 66.03, 0.7780, 33



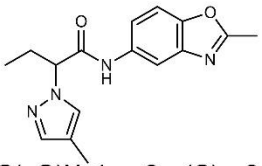
COc1cc(C#N)ccc1OC(C)c1nc2ccccc2o1

3.8481, 68.28, 0.7307, 33



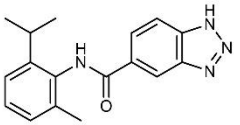
CC(NC(=O)Nc1ccccc1)C(=O)Nc1ccc(Cl)c1F

3.6278, 70.23, 0.7982, 37



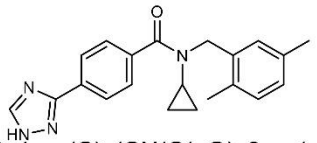
CCC(C(=O)Nc1ccc2oc(C)nc2c1)n1cc(C)cn1

3.2309, 72.95, 0.8022, 37



Cc1cccc(C(C)C)c1NC(=O)c1ccc2[nH]nnc2c1

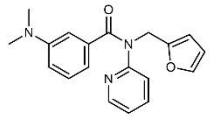
3.6420, 70.67, 0.7744, 35



Cc1ccc(C)c(CN(C(=O)c2ccc(-c3nc[nH]n3)cc2)C2CC2)c1

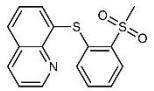
3.8933, 61.88, 0.7621, 46

Figure S3 Generated molecules with target condition #3 shown in Figure 7.



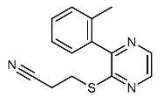
CN(C)c1cccc(C(=O)N(Cc2ccco2)c2ccccn2)c1

3.5876, 49.58, 0.7204, 39



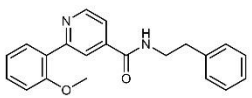
CS(=O)(=O)c1ccccc1Sc1cccc2cccnc12

3.7895 47.03 0.73756, 33



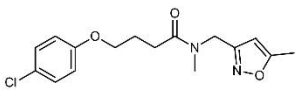
N#CCCSc1nccnc1-c1ccccc1C

3.4578, 49.57, 0.6198, 24



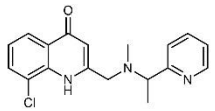
COc1ccccc1-c1cc(C(=O)NCCc2ccccc2)ccn1

3.7297, 51.22, 0.7482, 37



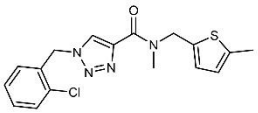
Cc1cc(CN(C)C(=O)CCCOc2ccc(Cl)cc2)no1

3.4540, 55.57, 0.7325, 35



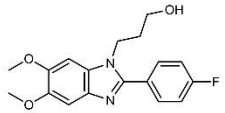
CC(c1ccccn1)N(C)Cc1cc(=O)c2cccc(Cl)c2[nH]1

3.7695, 48.99, 0.7932, 38



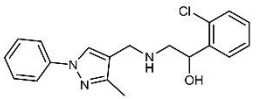
Cc1ccc(CN(C)C(=O)c2cn(Cc3ccccc3Cl)nn2)s1

3.6219, 51.02, 0.6978, 39



COc1cc2nc(-c3ccc(F)cc3)n(CCCO)c2cc1OC

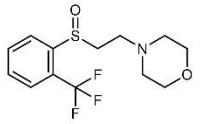
3.242, 56.51, 0.7537, 37



Cc1nn(-c2ccccc2)cc1CNCC(O)c1ccccc1Cl

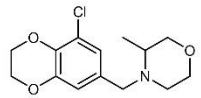
3.6573, 50.08, 0.7191, 35

Figure S4 Generated molecules with target condition #4 shown in Figure 7.



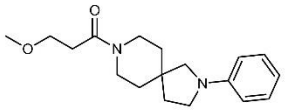
O=S(CCN1CCOCC1)c1ccccc1C(F)(F)F

2.1452, 29.54, 0.8535, 31



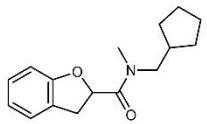
CC1COCCN1Cc1cc(Cl)c2c(c1)OCCO2

2.3319, 30.93, 0.8335, 29



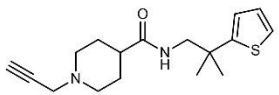
COCCC(=O)N1CCC2(CCN(c3ccccc3)C2)CC1

2.542, 32.78, 0.8567, 35



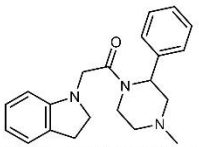
CN(CC1CCCC1)C(=O)C1Cc2ccccc2O1

2.6387, 29.54, 0.8349, 30



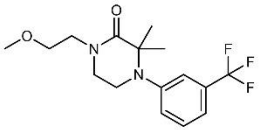
C#CCN1CCC(C(=O)NCC(C)(C)c2cccs2)CC1

2.4871, 32.34, 0.8478, 35



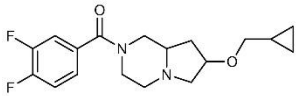
CN1CCN(C(=O)CN2CCc3ccccc32)C(c2ccccc2)C1

2.5644, 26.79, 0.8625, 40



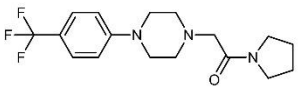
COCCN1CCN(c2cccc(C(F)(F)F)c2)C(C)(C)C1=O

2.7790, 32.78, 0.8510, 40



O=C(c1ccc(F)c(F)c1)N1CCN2CC(OCC3CC3)CC2C1

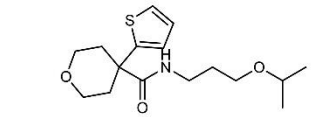
2.2901, 32.78, 0.8457, 41



O=C(CN1CCN(c2ccc(C(F)(F)F)cc2)CC1)N1CCCC1

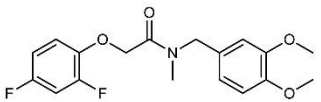
2.4498, 26.79, 0.8450, 41

Figure S5 Generated molecules with target condition #5 shown in Figure 7.



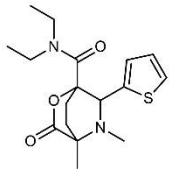
CC(C)OCCCNC(=O)C1(c2cccs2)CCOCC1

2.7276, 47.56, 0.7877, 32



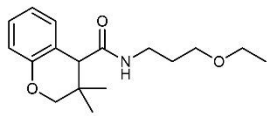
COc1ccc(CN(C)C(=O)COc2ccc(F)cc2F)cc1OC

3.0194, 48.00, 0.7694, 38



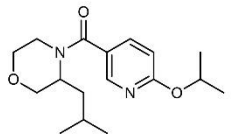
CCN(CC)C(=O)C12CCC(C)(C(=O)O1)N(C)C2c1cccs1

2.4375, 49.85, 0.7926, 43



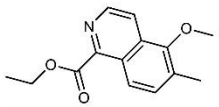
CCOCCCNC(=O)C1c2ccccc2OCC1(C)C

2.7316, 47.56, 0.8198, 30



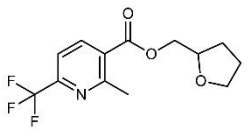
CC(C)CC1COCCN1C(=O)c1ccc(OC(C)C)nc1

2.7559, 51.66, 0.8391, 35



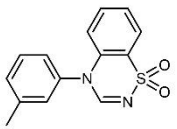
CCOC(=O)c1nccc2c(OC)c(C)ccc12

2.7285, 48.42, 0.7800, 29



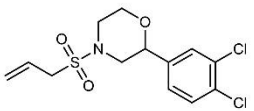
O=C(OCC1CCCO1)c1ccc(C(F)(F)F)nc1C

2.7446, 48.42, 0.8028, 33



Cc1cccc(N2C=NS(=O)(=O)c3ccccc32)c1

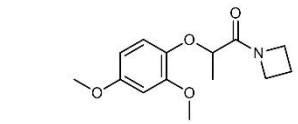
2.8637, 49.74, 0.8017, 34



C=CCS(=O)(=O)N1CCOC(c2ccc(Cl)c(Cl)c2)C1

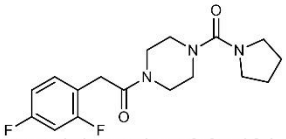
2.8825, 46.61, 0.7946, 37

Figure S6 Generated molecules with target condition #6 shown in Figure 7.



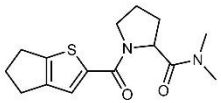
COc1ccc(OC(C)C(=O)N2CCC2)c(OC)c1

1.7034, 48.00, 0.8128, 32



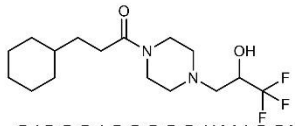
O=C(Cc1ccc(F)cc1F)N1CCN(C(=O)N2CCCC2)CC1

1.8673, 43.86, 0.8269, 40



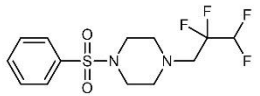
CN(C)C(=O)C1CCCN1C(=O)c1cc2c(s1)CCC2

1.9296, 40.62, 0.8355, 36



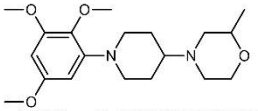
O=C(CCC1CCCCC1)N1CCN(CC(O)C(F)(F)F)CC1

2.4143, 43.78, 0.838463, 38



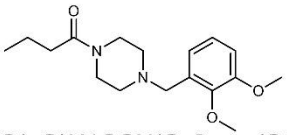
O=S(=O)(c1ccccc1)N1CCN(CC(F)(F)C(F)(F)F)CC1

1.8933, 40.62, 0.7689, 40



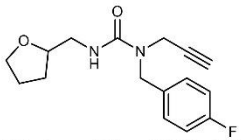
COc1cc(OC)cc(N2CCC(N3CCOC(C)C3)CC2)c1OC

2.4019, 43.40, 0.8125, 39



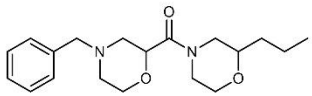
CCCC(=O)N1CCN(Cc2cccc(OC)c2OC)CC1

2.1481, 42.01, 0.8074, 33



C#CCN(Cc1ccc(F)cc1)C(=O)NCC1CCCO1

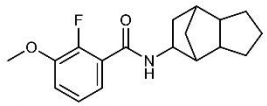
2.1495, 41.57, 0.8443, 33



CCCC1CN(C(=O)C2CN(Cc3ccccc3)CCO2)CCO1

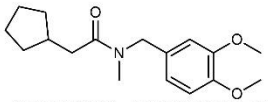
1.9149, 42.01, 0.8260, 37

Figure S7 Generated molecules with target condition #7 shown in Figure 7.



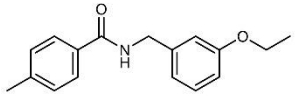
COc1cccc(C(=O)NC2CC3CC2C2CCCC32)c1F

3.3888, 38.33, 0.9293, 35



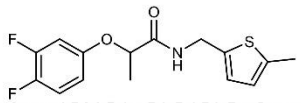
COc1ccc(CN(C)C(=O)CC2CCCC2)cc1OC

3.2425, 38.77, 0.8074, 32



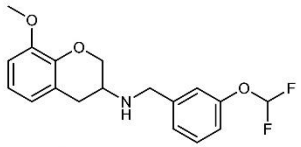
CCOc1cccc(CNC(=O)c2ccc(C)cc2)c1

3.3237, 38.33, 0.9039, 31



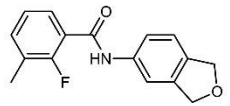
Cc1ccc(CNC(=O)C(C)Oc2ccc(F)c(F)c2)s1

3.4184, 38.33, 0.9186, 36



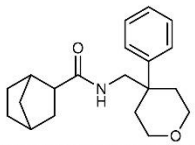
COc1cccc2c1OCC(NCc1cccc(OC(F)F)c1)C2

3.3899, 39.72, 0.8784, 36



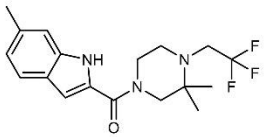
Cc1cccc(C(=O)Nc2ccc3c(c2)COC3)c1F

3.4166, 38.33, 0.9085, 33



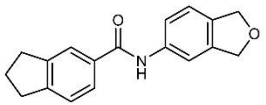
O=C(NCC1(c2ccccc2)CCOCC1)C1CC2CCC1C2

3.2872, 38.33, 0.9261, 36



Cc1ccc2cc(C(=O)N3CCN(CC(F)(F)F)C(C)(C)C3)[nH]c2c1

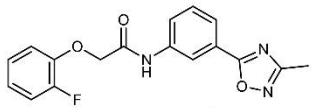
3.5750, 39.34, 0.8964, 46



O=C(Nc1ccc2c(c1)COC2)c1ccc2c(c1)CCC2

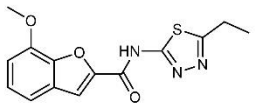
3.4578, 38.33, 0.9139, 36

Figure S8 Generated molecules with target condition #8 shown in Figure 7.



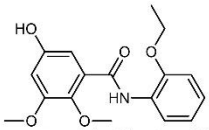
Cc1noc(-c2cccc(NC(=O)COc3ccccc3F)c2)n1

3.2016, 77.25, 0.7784, 38



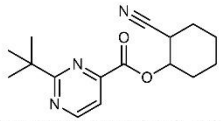
CCc1nnc(NC(=O)c2cc3cccc(OC)c3o2)s1

3.1076, 77.25, 0.8010, 34



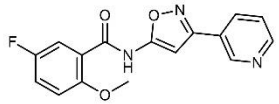
CCOc1ccccc1NC(=O)c1cc(O)cc(OC)c1OC

3.0604, 77.02, 0.8560, 34



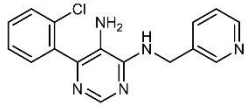
N#CC1CCCCC1OC(=O)c1ccnc(C(C)(C)C)n1

3.0133, 75.87, 0.7814, 35



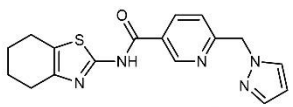
COc1ccc(F)cc1C(=O)Nc1cc(-c2cccnc2)no1

3.1366, 77.25, 0.8004, 37



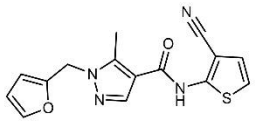
Nc1c(NCc2cccnc2)ncnc1-c1ccccc1Cl

3.3863, 76.72, 0.7721, 31



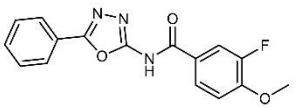
O=C(Nc1nc2c(s1)CCCC2)c1ccc(Cn2cccn2)nc1

2.9140, 72.7, 0.7931, 39



Cc1c(C(=O)Nc2sccc2C#N)cnn1Cc1ccco1

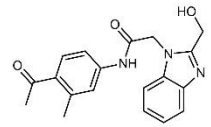
3.0183, 83.85, 0.8023, 34



COc1ccc(C(=O)Nc2nnc(-c3ccccc3)o2)cc1F

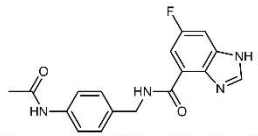
3.1366, 77.25, 0.8004, 37

Figure S9 Generated molecules with target condition #9 shown in Figure 7.



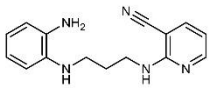
CC(=O)c1ccc(NC(=O)Cn2c(CO)nc3ccccc32)cc1C

2.6783, 84.22, 0.7012, 41



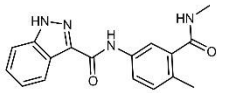
CC(=O)Nc1ccc(CNC(=O)c2cc(F)cc3[nH]cnc23)cc1

2.5904, 86.88, 0.6885, 40



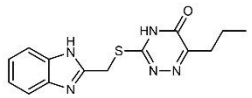
N#Cc1cccnc1NCCCNc1ccccc1N

2.4496, 86.76, 0.5525, 25



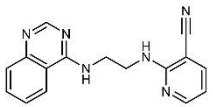
CNC(=O)c1cc(NC(=O)c2n[nH]c3ccccc23)ccc1C

2.4832, 86.88, 0.6944, 37



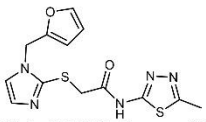
CCCc1nnc(SCc2nc3ccccc3[nH]2)[nH]c1=O

2.2860, 87.32, 0.7060, 30



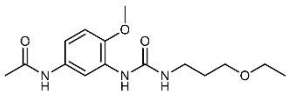
N#Cc1cccnc1NCCNc1ncnc2ccccc12

2.4205, 86.52, 0.7017, 29



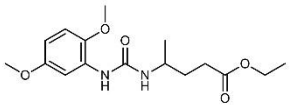
Cc1nnc(NC(=O)CSc2nccn2Cc2ccco2)s1

2.4151, 85.84, 0.6963, 33



CCOCCCNC(=O)Nc1cc(NC(C)=O)ccc1OC

2.2017, 88.69, 0.6423, 32



CCOC(=O)CCC(C)NC(=O)Nc1cc(OC)ccc1OC

2.5571, 85.89, 0.7178, 35

Figure S10 Generated molecules with target condition #10 shown in Figure 7.

Bibliography

[1] Liu, Z.-K.: MATERIALS GENOME, Inc. http://www.materialsgenome.com Accessed 2022-10-15

[2] Material development with AI (AIで材料開発), NIKKEI ELECTRONICS (2018. 10)

[3] Sanchez-Lengeling, B., Aspuru-Guzik, A.: Inverse molecular design using machine learning: Generative models for matter engineering. ***Science*** 361(6400), 360–365 (2018)

[4] Kim, H., Na, J., Lee, W.B.: Generative chemical transformer: Neural machine learning of molecular geometric structures from chemical language via attention. ***Journal of Chemical Information and Modeling*** 61(12), 5804–5814 (2021)

[5] Rogers, D., Hahn, M.: Extended-connectivity fingerprints. ***Journal of chemical information and modeling*** 50(5), 742–754 (2010)

[6] Morgan, H.L.: 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)

[7] Polykovskiy, D., Zhebrak, A., Sanchez-Lengeling, B., Golovanov, S., Tatanov, O., Belyaev, S., Kurbanov, R., Artamonov, A., Aladinskiy, V., Veselov, M., et al.: Molecular sets (MOSES): a benchmarking platform for molecular generation models. ***Frontiers in pharmacology*** 11, 565644 (2020)

[8] Moleculenet. https://moleculenet.org/Accessed 2022-11-16

[9] SMART-Language for Describing Molecular Patterns. https://www.daylight.com/dayhtml/doc/theory/theory.smarts.html Accessed 2021-06-04

[10] Heller, S.R., McNaught, A., Pletnev, I., Stein, S., Tchekhovskoi, D.: Inchi, the IUPAC international chemical identifier**.** ***Journal of cheminformatics*** 7(1), 1–34 (2015)

[11] Weininger, D.: 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)

[12] Weininger, D., Weininger, A., Weininger, J.L.: Smiles. 2. algorithm for generation of unique smiles notation. ***Journal of Chemical Information and Computer Sciences*** 29(2), 97–101 (1989)

[13] Weininger, D.: Smiles. 3. depict. graphical depiction of chemical structures. ***Journal of Chemical Information and Computer Sciences*** 30(3), 237–243 (1990)

[14] Ar´us-Pous, J., Johansson, S.V., Prykhodko, O., Bjerrum, E.J., Tyrchan, C., Reymond, J.-L., Chen, H., Engkvist, O.: Randomized smiles strings improve the quality of molecular generative models. ***Journal of cheminformatics*** 11(1), 1–13 (2019)

[15] Amabilino, S., Pog´any, P., Pickett, S.D., Green, D.V.S.: Guidelines for recurrent neural network transfer learning-based molecular generation of focused libraries. ***Journal of Chemical Information and Modeling*** 60(12), 5699–5713 (2020)

[16] Segler, M.H., Kogej, T., Tyrchan, C., Waller, M.P.: Generating focused molecule libraries for drug discovery with recurrent neural networks. ***ACS central science*** 4(1), 120–131 (2018)

[17] Winter, R., Montanari, F., No´e, F., Clevert, D.-A.: Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations. ***Chemical science*** 10(6), 1692–1701 (2019)

[18] Gao, K., Nguyen, D.D., Tu, M., Wei, G.-W.: Generative network complex for the automated generation of drug-like molecules. ***Journal of chemical information and modeling*** 60(12), 5682–5698 (2020)

[19] Kingma, D.P., Welling, M.: Auto-encoding variational Bayes. ***arXiv preprint*** arXiv:1312.6114 (2013)

[20] G´omez-Bombarelli, R., Wei, J.N., Duvenaud, D., Hern´andez-Lobato, J.M., S´anchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T.D., Adams, R.P., Aspuru-Guzik, A.: Automatic chemical design using a data-driven continuous representation of molecules. ***ACS central science*** 4(2), 268–276 (2018)

[21] Blaschke, T., Olivecrona, M., Engkvist, O., Bajorath, J., Chen, H.: Application of generative autoencoder in de novo molecular design. ***Molecular informatics*** 37(1-2), 1700123 (2018)

[22] Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. ***arXiv preprint*** arXiv:1406.2661 (2014)

[23] Prykhodko, O., Johansson, S.V., Kotsias, P.-C., Ar´us-Pous, J., Bjerrum, E.J., Engkvist, O., Chen, H.: A de novo molecular generation method using latent vector based generative adversarial network. ***Journal of Cheminformatics*** 11(1), 1–13 (2019)

[24] Olivecrona, M., Blaschke, T., Engkvist, O., Chen, H.: Molecular de-novo design through deep reinforcement learning. ***Journal of cheminformatics*** 9(1), 1–14 (2017)

[25] Griffiths, R.-R., Hern´andez-Lobato, J.M.: Constrained bayesian optimization for automatic chemical design using variational autoencoders. ***Chemical science*** 11(2), 577–586 (2020)

[26] Winter, R., Montanari, F., Steffen, A., Briem, H., No´e, F., Clevert, D. A.: Efficient multi-objective molecular optimization in a continuous latent space. ***Chemical science*** 10(34), 8016–8024 (2019)

[27] Janet, J.P., Chan, L., Kulik, H.J.: Accelerating chemical discovery with machine learning: simulated evolution of spin crossover complexes with an artificial neural network. ***The Journal of Physical Chemistry Letters*** 9(5), 1064–1071 (2018)

[28] Moosavi, S.M., Chidambaram, A., Talirz, L., Haranczyk, M., Stylianou, K.C., Smit, B.: Capturing chemical intuition in synthesis of metal-organic frameworks. ***Nature communications*** 10(1), 1–7 (2019)

[29] Nigam, A., Friederich, P., Krenn, M., Aspuru-Guzik, A.: Augmenting genetic algorithms with deep neural networks for exploring the chemical space. ***arXiv preprint*** arXiv:1909.11655 (2019)

[30] Yang, X., Zhang, J., Yoshizoe, K., Terayama, K., Tsuda, K.: Chemts: an efficient python library for de novo molecular generation. ***Science and technology of advanced materials*** 18(1), 972–976 (2017)

[31] Segler, M.H., Preuss, M., Waller, M.P.: Planning chemical syntheses with deep neural networks and symbolic AI. ***Nature*** 555(7698), 604–610 (2018)

[32] Zhang, X., Zhang, K., Lee, Y.: Machine learning enabled tailor-made design of application-specific metal–organic frameworks. ***ACS applied materials & interfaces*** 12(1), 734–743 (2019)

[33] Kotsias, P.-C., Ar´us-Pous, J., Chen, H., Engkvist, O., Tyrchan, C., Bjerrum, E.J.: Direct steering of de novo molecular generation with descriptor conditional recurrent neural networks. ***Nature Machine Intelligence*** 2(5), 254–265 (2020)

[34] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. ***Advances in neural information processing systems*** 30 (2017)

[35] Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. ***arXiv preprint*** arXiv:1409.0473 (2014)

[36] Li, X., Fourches, D.: Smiles pair encoding: a data-driven substructure tokenization algorithm for deep learning. ***Journal of Chemical Information and Modeling*** 61(4), 1560–1569 (2021)

[37] Sohn, K., Lee, H., Yan, X.: Learning structured output representation using deep conditional generative models. ***Advances in neural information processing systems*** 28 (2015)

[38] Xiong, R., Yang, Y., He, D., Zheng, K., Zheng, S., Xing, C., Zhang, H., Lan, Y., Wang, L., Liu, T.: On layer normalization in the transformer architecture. In: ***International Conference on Machine Learning***, pp. 10524–10533 (2020)

[39] He, J., Spokoyny, D., Neubig, G., Berg-Kirkpatrick, T.: Lagging inference networks and posterior collapse in variational autoencoders. ***arXiv preprint*** arXiv:1901.05534 (2019)

[40] Bowman, S.R., Vilnis, L., Vinyals, O., Dai, A.M., Jozefowicz, R., Bengio, S.: Generating sentences from a continuous space. ***arXiv preprint*** arXiv:1511.06349 (2015)

[41] Wildman, S.A., Crippen, G.M.: Prediction of physicochemical parameters by atomic contributions. ***Journal of chemical information and computer sciences*** 39(5), 868–873 (1999)

[42] Ertl, P., Rohde, B., Selzer, P.: Fast calculation of molecular polar surface area as a sum of fragment-based contributions and its application to the prediction of drug transport properties. ***Journal of medicinal chemistry*** 43(20), 3714–3717 (2000)

[43] Bickerton, G.R., Paolini, G.V., Besnard, J., Muresan, S., Hopkins, A.L.: Quantifying the chemical beauty of drugs. ***Nature chemistry*** 4(2), 90–98 (2012)

[44] Reddy, D.R.: Speech understanding systems: A summary of results of the five-year research effort. department of computer science. Carnegie-Mellon University, Pittsburgh, PA 17 (1977)

[45] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. ***The journal of machine learning research*** 15(1), 1929–1958 (2014)

[46] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. ***arXiv preprint*** arXiv:1412.6980 (2014)

[47] Loshchilov, I., Hutter, F.: SGDR: Stochastic gradient descent with warm restarts. ***arXiv preprint*** arXiv:1608.03983 (2016)

[48] Sterling, T., Irwin, J.J.: Zinc 15–ligand discovery for everyone. ***Journal of chemical information and modeling*** 55(11), 2324–2337 (2015)

[49] Landrum, G.: Rdkit: open-source cheminformatics from machine learning to chemical registration. ***In: Abstracts of Papers of the American Chemical Society***, 258 (2019)

[50] Preuer, K., Renz, P., Unterthiner, T., Hochreiter, S., Klambauer, G.: Fr´echet chemnet distance: a metric for generative models for molecules in drug discovery. ***Journal of chemical information and modeling*** 58(9), 1736–1741 (2018)

[51] Tanimoto, T.T.: Elementary mathematical theory of classification and prediction (1958)

[52] Degen, J., Wegscheid-Gerlach, C., Zaliani, A., Rarey, M.: On the art of compiling and using' drug-like' chemical fragment spaces. ***ChemMedChem*** 3(10), 1503–1507 (2008)

[53] Bemis, G.W., Murcko, M.A.: The properties of known drugs. 1. Molecular frameworks. ***Journal of medicinal chemistry*** 39(15), 2887–2893 (1996)

[54] Dollar, O., Joshi, N., Beck, D.A., Pfaendtner, J.: Attention-based generative models for de novo molecular design. ***Chemical Science*** 12(24), 8362–8372 (2021)

[55] Jin, W., Barzilay, R., Jaakkola, T.: Junction tree variational autoencoder for molecular graph generation. ***In: International Conference on Machine Learning***, pp. 2323–2332 (2018). PMLR

[56] Vig, J.: A multiscale visualization of attention in the transformer model. ***arXiv preprint*** arXiv:1906.05714 (2019)

[57] Fu, H., Li, C., Liu, X., Gao, J., Celikyilmaz, A., Carin, L.: Cyclical annealing schedule: A simple approach to mitigating KL vanishing. ***arXiv preprint*** arXiv:1903.10145 (2019)

[58] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. ***arXiv preprint*** arXiv:1810.04805 (2018)

[59] Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H.P.d.O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., et al.: Evaluating large language models trained on code. ***arXiv preprint*** arXiv:2107.03374 (2021)

[60] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. ***Journal of Machine Learning Research*** 21(140), 1–67 (2020)

[61] Greenaway, R.L., Jelfs, K.E.: Integrating computational and experimental workflows for accelerated organic materials discovery. ***Advanced Materials*** 33(11), 2004831 (2021)

[62] Sylvinson MR, D., Chen, H.-F., Martin, L.M., Saris, P.J., Thompson, M.E.: Rapid multiscale computational screening for OLED host materials. ***ACS applied materials & interfaces*** 11(5), 5276–5288 (2019)

[63] Kim, K., Kang, S., Yoo, J., Kwon, Y., Nam, Y., Lee, D., Kim, I., Choi, Y.S., Jung, Y., Kim, S., et al.: Deep-learning-based inverse design model for intelligent discovery of organic molecules. ***npj Computational Materials*** 4(1), 1–7 (2018)

[64] Zunger, A.: Inverse design in search of materials with target functionalities. ***Nature Reviews Chemistry*** 2(4), 1–16 (2018)

[65] Lyu, M.-Y., Choi, T.G.: Research trends in polymer materials for use in lightweight vehicles. ***International journal of precision engineering and manufacturing*** 16(1), 213–220 (2015)

[66] Stauber, R.: Plastics in automotive engineering. ***ATZ worldwide*** 109(3), 2–4 (2007)

[67] Furka, A.: Study on the possibilities of systematic searching for pharmaceutically useful peptides. Notarized Report (File number 36237/1982, in Hungarian) (1982)

[68] Gottipati, S.K., Sattarov, B., Niu, S., Pathak, Y., Wei, H., Liu, S., Blackburn, S., Thomas, K., Coley, C., Tang, J., et al.: Learning to navigate the synthetically accessible chemical space using reinforcement learning. ***In Proceedings of International Conference on Machine Learning***, pp. 3668–3679 (2020)

[69] Yang, S., Hwang, D., Lee, S., Ryu, S., Hwang, S.J.: Hit and lead discovery with explorative RL and fragment-based molecule generation. ***Advances in Neural Information Processing Systems*** 34, 7924–7936 (2021)

[70] SV, S.S., Law, J.N., Tripp, C.E., Duplyakin, D., Skordilis, E., Biagioni, D., Paton, R.S., St John, P.C.: Multi-objective goal-directed optimization of de novo stable organic radicals for aqueous redox flow batteries. ***Nature Machine Intelligence*** 4(8), 720–730 (2022)

[71] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. ***arXiv preprint*** arXiv:1707.06347 (2017)

[72] Mendez, D., Gaulton, A., Bento, A.P., Chambers, J., De Veij, M., F´elix, E., Magarı̃nos, M.P., Mosquera, J.F., Mutowo, P., Nowotka, M., et al.: Chembl: towards direct deposition of bioassay data. ***Nucleic acids research*** 47(D1), 930–940 (2019)

[73] Davies, M., Nowotka, M., Papadatos, G., Dedman, N., Gaulton, A., Atkinson, F., Bellis, L., Overington, J.P.: Chembl web services: streamlining access to drug discovery data and utilities. ***Nucleic acids research*** 43(W1), 612–620 (2015)

[74] PubChem: PubChem SARS-Cov-2 Clinical Trials. https://pubchem.ncbi.nlm.nih.gov/#tab=compound&query=covid-19%20clinicaltrials Accessed 2022-05-06

[75] Huang, C.-C., Yeh, C.-M., Wu, M.-Y., Hsu, K.-S.: A single in vivo cocaine administration impairs 5-ht 1b receptor-induced long-term depression in the nucleus accumbens. ***Journal of neurochemistry*** 125(6), 809–821 (2013)

[76] Cai, X., Kallarackal, A.J., Kvarta, M.D., Goluskin, S., Gaylor, K., Bailey, A.M., Lee, H.-K., Huganir, R.L., Thompson, S.M.: Local potentiation of excitatory synapses by serotonin and its alteration in rodent models of depression. ***Nature Neuroscience*** 16(4), 464–472 (2013)

[77] Shoval, G., Bar-Shira, O., Zalsman, G., Mann, J.J., Chechik, G.: Transitions in the transcriptome of the serotonergic and dopaminergic systems in the human brain during adolescence. European ***Neuropsychopharmacology*** 24(7), 1123–1132 (2014)

[78] Tepper, S.J., Rapoport, A.M., Sheftell, F.D.: Mechanisms of Action of the 5-HT1B/1D Receptor Agonists. ***Archives of Neurology*** 59(7), 1084–1088 (2002)

[79] Yadav, V.K., Ryu, J.-H., Suda, N., Tanaka, K.F., Gingrich, J.A., Sch¨utz, G., Glorieux, F.H., Chiang, C.Y., Zajac, J.D., Insogna, K.L., Mann, J.J., Hen, R., Ducy, P., Karsenty, G.: Lrp5 controls bone formation by inhibiting serotonin synthesis in the duodenum. ***Cell*** 135(5), 825–837 (2008)

[80] Blackburn, T.P.: Serotonin (5-hydroxytryptamine; 5-ht): Receptors. In: Squire, L.R. (ed.) Encyclopedia of Neuroscience, pp. 701–714. Academic Press, Oxford (2009)

[81] Tiger, M., Varn¨as, K., Okubo, Y., Lundberg, J.: The 5-ht1b receptor a potential target for antidepressant treatment. ***Psychopharmacology*** 235(5), 1317–1334 (2018)

[82] Yang, P., Yang, M., Li, P., Cao, D., Gong, D., Lv, J., Pu, L., Huang, S., Liang, Y.: A meta-analysis of 5-hydroxytryptamine receptor 1b polymorphisms with risk of major depressive disorder and suicidal behavior. ***Frontiers in psychiatry*** 12 (2021)

[83] Nilsson, T., Longmore, J., Shaw, D., Olesen, I.J., Edvinsson, L.: Contractile 5-ht1b receptors in human cerebral arteries: pharmacological characterization and localization with immunocytochemistry. ***British journal of pharmacology*** 128(6), 1133–1140 (1999)

[84] Hansen, H.D., Mandeville, J.B., Sander, C.Y., Hooker, J.M., Catana, C., Rosen, B.R., Knudsen, G.M.: Functional characterization of 5-ht1b receptor drugs in nonhuman primates using simultaneous pet-mr. ***Journal of Neuroscience*** 37(44), 10671–10678 (2017)

[85] Alhossary, A., Handoko, S.D., Mu, Y., Kwoh, C.-K.: Fast, accurate, and reliable molecular docking with quickvina 2. ***Bioinformatics*** 31(13), 2214–2216 (2015)

[86] Clark, M., Neumaier, J.: The 5-ht1b receptor: behavioral implications. ***Psychopharmacology bulletin*** 35(4), 170–185 (2001)

[87] Morikawa, H., Manzoni, O.J., Crabbe, J.C., Williams, J.T.: Regulation of central synaptic transmission by 5-ht1bauto-and heteroreceptors. ***Molecular pharmacology*** 58(6), 1271–1278 (2000)

[88] Yang, Y., Zhang, L., Yu, J., Ma, Z., Li, M., Wang, J., Hu, P., Zou, J., Liu, X., Liu, Y., et al.: A novel 5-ht1b receptor agonist of herbal compounds and one of the therapeutic uses for alzheimer's disease. ***Frontiers in pharmacology***, 2362 (2021)

[89] Pauwels, P.J.: 5-ht1bd receptor antagonists. ***General Pharmacology: The Vascular System*** 29(3), 293–303 (1997)

[90] Hasuo, H., Matsuoka, T., Akasu, T.: Activation of presynaptic 5-hydroxytryptamine 2a receptors facilitates excitatory synaptic transmission via protein kinase c in the dorsolateral septal nucleus. ***Journal of Neuroscience*** 22(17), 7509–7517 (2002)

[91] Dai, S., Yu, L., Shi, X., Wu, H., Shao, P., Yin, G., Wei, Y.: Serotonin regulates osteoblast proliferation and function in vitro. Brazilian Journal of Medical and ***Biological Research*** 47, 759–765 (2014)

[92] Kristjansdottir, H.L., Lewerin, C., Lerner, U.H., Waern, E., Johansson, H., Sundh, D., Karlsson, M., Cummings, S.R., Zetterberg, H., Lorentzon, M., et al.: High serum serotonin predicts increased risk for hip fracture and nonvertebral osteoporotic fractures: the mros Sweden study**.** ***Journal of Bone and Mineral Research*** 33(9), 1560–1567 (2018)

[93] de Vernejoul, M.-C., Collet, C., Chabbi-Achengli, Y.: Serotonin: good or bad for bone. ***BoneKEy reports*** 1 (2012)

[94] Chabbi-Achengli, Y., Coudert, A.E., Callebert, J., Geoffroy, V., Cˆot´e, F., Collet, C., de Vernejoul, M.-C.: Decreased osteoclastogenesis in serotonin-deficient mice. ***Proceedings of the National Academy of Sciences*** 109(7), 2567–2572 (2012)

[95] Collet, C., Schiltz, C., Geoffroy, V., Maroteaux, L., Launay, J.-M., De Vernejoul, M.-C.: The serotonin 5-ht2b receptor controls bone mass via osteoblast recruitment and proliferation. ***The FASEB Journal*** 22(2), 418 (2008)

[96] Asenjo, J., Herrera, L., Byrne, B.: Development of an expert system for selection and synthesis of protein purification processes. ***Journal of Biotechnology*** 11(2-3), 275–298 (1989)

[97] Kempf, K.G.: Manufacturing and artificial intelligence. ***Robotics*** 1(1), 13–25 (1985)

[98] Stephanopoulos, G.: Artificial intelligence in process engineering—current state and future trends. ***Computers & Chemical Engineering*** 14(11), 1259–1270 (1990)

[99] Beltramini, L., Motard, R.: Knod—a knowledge based approach for process design. ***Computers & Chemical Engineering*** 12(9-10), 939–958 (1988)

[100] Kirkwood, R.L., Locke, M.H., Douglas, J.: A prototype expert system for synthesizing chemical process flowsheets. ***Computers & chemical engineering*** 12(4), 329–343 (1988)

[101] Fang, H., Zhou, J., Wang, Z., Qiu, Z., Sun, Y., Lin, Y., Chen, K., Zhou, X., Pan, M.: Hybrid method integrating machine learning and particle swarm optimization for smart chemical process operations. ***Frontiers of Chemical Science and Engineering*** 16(2), 274–287 (2022)

[102] Zhang, Z., Wu, Z., Rincon, D., Christofides, P.D.: Real-time optimization and control of nonlinear processes using machine learning. ***Mathematics*** 7(10), 890 (2019)

[103] Liang, F., Zhang, T.: Application of neural network in optimization of chemical process. ***arXiv preprint*** arXiv:2110.04942 (2021)

[104] Park, S., Shim, J., Yoon, N., Lee, S., Kwak, D., Lee, S., Kim, Y.M., Son, M., Cho, K.H.: Deep reinforcement learning in an ultrafiltration system: Optimizing operating pressure and chemical cleaning conditions. ***Chemosphere*** 308, 136364 (2022)

[105] Lee, J., Hong, S., Cho, H., Lyu, B., Kim, M., Kim, J., Moon, I.: Machine learning-based energy optimization for on-site smr hydrogen production. ***Energy Conversion and Management*** 244, 114438 (2021)

[106] Greenhill, S., Rana, S., Gupta, S., Vellanki, P., Venkatesh, S.: Bayesian optimization for adaptive experimental design: a review. ***IEEE Access*** 8, 13937–13948 (2020)

[107] Roch, L.M., H¨ase, F., Kreisbeck, C., Tamayo-Mendoza, T., Yunker, L.P., Hein, J.E., Aspuru-Guzik, A.: Chemos: orchestrating autonomous experimentation. ***Science Robotics*** 3(19), 5559 (2018)

[108] Roch, L.M., H¨ase, F., Kreisbeck, C., Tamayo-Mendoza, T., Yunker, L.P., Hein, J.E., Aspuru-Guzik, A.: Chemos: An orchestration software to democratize autonomous discovery. ***PLoS One*** 15(4), 0229862 (2020)

[109] Coley, C.W., Thomas III, D.A., Lummiss, J.A., Jaworski, J.N., Breen, C.P., Schultz, V., Hart, T., Fishman, J.S., Rogers, L., Gao, H., et al.: A robotic platform for flow synthesis of organic compounds informed by AI planning. ***Science*** 365(6453), 1566 (2019)

Abstract in Korean

국문 초록

딥러닝을 활용한 자율적 물질 발굴: 목표지향적 분자 역설계

김 현 승

화학생물공학부

서울대학교 대학원

청동의 발견이 석기 시대의 종지부를 찍었듯이, 인간 문명의 진보는 더 나은 물질을 발굴하는 것과 밀접한 연관이 있다. 현대의 물질 발굴은 전기 전자 소재, 에너지 물질, 세라믹, 촉매, 나노 물질, 바이오 물질, 그리고 바이오 물질에 이르기까지 여러 분야에 걸쳐 있다. 이러한 물질 개발 연구들에 있어, 화학물질이 존재하는 공간은 매우 방대하기에 효율적인 탐색을 통해 원하는 물질을 찾아내는 것은 매우 도전적인 일이다. 과거에는 전문가의 지식이나 직관에 의존해 원하는 물질을 탐색했으나 이는 방대한 공간을 효율적으로 빠르게 탐색하기에는 효과적이지 못하다. 때문에 물질 개발 프로세스 자체를 하나의 closed-loop system으로 구성하고 이를 AI/ML이 운전함으로써 보다 빠르고 효율적으로 원하는 물질을 개발하려는 시도가 이어지고 있다. AI/ML에 의해 운전되는 closed-loop 물질 발굴 system은 물질 역설계, 물질 scoring,

반응경로 합성, and process construction 등의 모듈간 상호작용으로 이루어진다.

본 연구에서는 위에서 열거한 모듈들 중 분자 역설계 모듈에 대해서 두 가지 모델을 다룬다. 하나는 화학 언어를 사용하는 목표 지향 역분자 설계 모델로, 목표하는 물성들을 동시에 만족하는 자연스러운 분자 역설계가 가능하다. 이 모델은 조건부 생성모델에 Transformer라는 자연어 처리 모델을 내장한 구조를 갖는데, 분자 구조를 나타내는 언어적 순서의 패턴을 인식하여 화학 규칙을 만족하면서도 실제 분자와 유사한 가상의 분자를 설계한다. 다른 하나는 극단적인 물성을 만족하는 분자 설계를 위해 combinatorial chemistry와 강화학습을 활용한 모델이다. Neural machine translator, variational autoencoder, 그리고 generative adversarial network에 기반한 기존의 확률 분포 학습 모델에 근거한 분자 역설계 모델들은 알려진 물질 (훈련 데이터) 분포를 근사하는 학습 모델을 도출하기에 알려진 물질의 분포를 벗어난 (극단적인 물성을 가진) 물질을 생성하지 못한다. 그러나 제안된 모델은 combinatorial chemistry가 분자 조각 조합에서 나타날 수 있는 모든 물성을 가진 분자를 생성할 수 있다는 점과 강화학습 기법이 목표 물성을 만족하는 분자 조각 선택 정책을 학습시킬 수 있다는 점을 결합하여 극단적인 물성을 만족하는 분자 역설계가 가능하다.

제안된 모델들은 sub-second 안에 목표하는 분자 구조를 역설계가 가능하다. 이런 점을 고려할 때 AI/ML을 활용한 물질 발굴 기법들이 물질 개발 프로세스를 가속화하는 데 크게 기여할 수 있을 것으로

기대한다.

주요어: 물질 발굴, 목표지향적 분자 역설계, 화학 언어, 분자 조각 기반 분자 역설계, 인공지능에 기반한 자율적 조합 화학 분자 역설계

학 번: 2018-20184