Contents lists available at ScienceDirect

Computers in Biology and Medicine

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

ReBADD-SE: Multi-objective molecular optimisation using SELFIES fragment and off-policy self-critical sequence training

Jonghwan Choi [a,c], Sangmin Seo [a,c], Seungyeon Choi [a], Shengmin Piao [a], Chihyun Park [b,c], Sung Jin Ryu [c], Byung Ju Kim [c], Sanghyun Park [a,*]

[a] Department of Computer Science, Yonsei University, Yonsei-ro 50, Seodaemun-gu, 03722, Seoul, Republic of Korea
[b] Department of Computer Science and Engineering, Kangwon National University, Chuncheon-si, 24341, Kangwon-do, Republic of Korea
[c] UBLBio Corporation, Yeongtong-ro 237, Suwon, 16679, Gyeonggi-do, Republic of Korea

ARTICLE INFO

Dataset link: https://github.com/mathcom/ReBADD-SE

Keywords:
Drug discovery
De novo drug design
Multi-objective optimisation
SELFIES
Reinforcement learning

ABSTRACT

The discovery of drugs to selectively remove disease-related cells is challenging in computer-aided drug design. Many studies have proposed multi-objective molecular generation methods and demonstrated their superiority using the public benchmark dataset for kinase inhibitor generation tasks. However, the dataset does not contain many molecules that violate Lipinski's rule of five. Thus, it remains unclear whether existing methods are effective in generating molecules violating the rule, such as navitoclax. To address this, we analysed the limitations of existing methods and propose a multi-objective molecular generation method with a novel parsing algorithm for molecular string representation and a modified reinforcement learning method for the efficient training of multi-objective molecular optimisation. The proposed model had success rates of 84% in GSK3b+JNK3 inhibitor generation and 99% in Bcl-2 family inhibitor generation tasks.

1. Introduction

The discovery of drug candidates that can selectively remove disease-related cells is crucial in computer-aided drug design research. There are 10^{30} to 10^{60} known drug-like compounds [1]. However, the number of drugs that are clinically available for complex diseases, such as rare cancers and age-related diseases, is limited [2,3]. These complex diseases are generally associated with more than one protein. Thus, drug candidates should be able to interact with multiple proteins related to diseases of interest, and researchers have been studying methods to design molecules with multiple objectives.

Many studies have proposed machine learning-based approaches with various molecular representation methods to overcome the difficulty of multi-objective drug design (Fig. 1) [4–12]. Generative tensorial reinforcement learning (GENTRL) is the most representative model for drug discovery; in one study, the used GENTRL to identify potential discoidin domain receptor 1 (DDR1) kinase inhibitors for the treatment of renal fibrosis [5]. GENTRL exploited molecular data formatted in the simplified molecular-input line-entry system (SMILES) and variational autoencoder (VAE) to efficiently identify many drug-like compounds and used reinforcement learning (RL) to find optimal molecular structures that can potently inhibit DDR1 kinase. Reinforcement Learning for Structural Evolution (ReLeaSE) also exploits SMILES and RL techniques, and it has demonstrated the effectiveness of the SMILES and RL combination by identifying desired compounds with maximal, minimal, or specific ranges of physical properties, such as melting point or hydrophobicity (logP) [4]. The molecular swarm optimiser (MSO) model was designed to identify candidates for epidermal growth factor receptor (EGFR) and beta-secretase 1 (BACE1) inhibitors using evolutionary optimisation techniques on SMILES [6]. MSO exploits VAE to generate a chemical latent space and applies a particle swarm optimisation algorithm to explore optimal points where VAE produces molecules with the desired binding affinity values against EGFR and BACE1 proteins.

The SMILES-based approach exhibited good performance in multi-objective optimisation. However, the sophisticated modification of molecular structures, such as molecular scaffold permutations, and providing explicit structural information to generative models are difficult [13]. To address this weakness of SMILES-based approaches, RationaleRL was proposed by combining RL and molecular graph representations, instead of SMILES [8]. To accomplish multi-objective molecular optimisation, for each objective property, RationaleRL first extracts property-related substructures from numerous molecular graphs and then generates molecules with the desired properties by reassembling optimal pieces among the extracted substructures. RationaleRL has demonstrated its superiority in a task where the goal was to design a molecule with improved inhibitory activities against glycogen synthase

* Corresponding author.
E-mail addresses: mathcombio@yonsei.ac.kr (J. Choi), sanghyun@yonsei.ac.kr (S. Park).

https://doi.org/10.1016/j.compbiomed.2023.106721
Received 20 December 2022; Received in revised form 11 February 2023; Accepted 26 February 2023
Available online 28 February 2023
0010-4825/© 2023 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

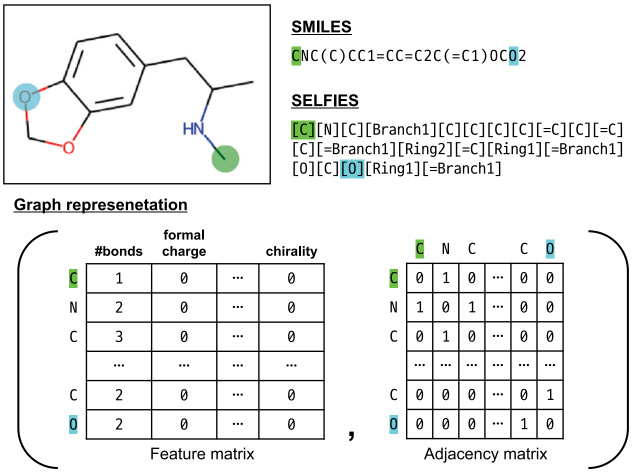
SMILES
CNC(C)CC1=CC=C2C(=C1)OCO2
SELFIES
[C][N][C][Branch1][C][C][C][C][=C][C][=C][C][=Branch1][Ring2][=C][Ring1][=Branch1][O][C][O][Ring1][=Branch1]
O
O
HN
Graph represenetation
#bonds
formal charge
chirality
C N C C O
Feature matrix
Adjacency matrix

Fig. 1. Molecular representations widely used in drug design.

kinase 3 beta (GSK3b) and c-Jun N-terminal kinase 3 (JNK3) proteins, while retaining high levels of synthetic accessibility (SA) and quantitative estimate of drug-likeness (QED). Other studies, such as Markov molecular sampling (MARS) and MolSearch, utilised Markov chain Monte Carlo sampling and Monte Carlo tree search methods instead of RL in multi-objective drug discovery tasks [9,12]. An optimal molecular structure was discovered by iteratively editing a seed molecule. These search-based models exhibited better performance than RationaleRL on the benchmark dataset for GSK3b, JNK3, QED, and SA optimisations.

Existing SMILES and graph-based models are superior in GSK3b and JNK3 dual inhibitor generation tasks. However, whether they are also suitable for generating more complicated drug candidates than GSK3b and JNK3 inhibitors is unclear. Many drugs, including GSK3b and JNK3 inhibitors, satisfy Lipinski's rule of five (Ro5), in which an orally active drug should pass at least three filters with the following criteria: a molecular weight (MWT) < 500 Da, no more than five hydrogen bond donors, no more than 10 hydrogen bond acceptors, and a calculated octanol-water partition coefficient (logP) < 5 [14]. In practice, however, a number of drugs lie outside the ranges specified in Ro5 [15] (Fig. A.1a). For example, navitoclax (also termed ABT-263) has a MWT of 975 Da and 14 hydrogen bond acceptors. Navitoclax is a potent Bcl-2 family protein inhibitor that can be used to activate intrinsic apoptosis of senescent and cancer cells [16–18]. However, because of side effects that include thrombocytopenia, its dose and efficacy are limited [16]. There is another Bcl-2 family inhibitor similar to navitoclax, named ABT-737, which is also an Ro5 outlier (MWT = 813 and logP > 7). ABT-737 has received attention as a potent senolytic drug because it significantly reduces the viability of DNA-damage-induced, oncogene-induced, or perforin-knockout senescence than that of non-senescent cells by inhibiting three Bcl-2 family proteins (Bcl-2, Bcl-xl, and Bcl-w) [17,19]. However, because ABT-737 is not orally bioavailable, discovery of new Bcl-2 family inhibitors that can simultaneously inhibit Bcl-2, Bcl-xl, and Bcl-w is both important and challenging.

Existing multi-objective molecular optimisation models can be exploited to generate Ro5 outlier molecules, such as navitoclax and ABT-737. However, SMILES-based generative models struggle against the complex grammar of SMILES owing to the difficulty in producing chemically valid molecular structures. Graph-based generative models have difficulty in learning large and heavy molecules because of the high computational and space costs caused by data preprocessing parts, such as graph decomposition algorithms [20]. Thus, developing a novel molecular generative model is necessary to efficiently generate Ro5 outlier molecules.

Self-referencing embedded strings (SELFIES) can be an alternative molecular representation method for generating molecules violating Ro5. SELFIES is an advanced molecular string representation because it always guarantees that every combination of SELFIES symbols corresponds to a chemically valid molecule [21], which is a considerable advantage compared to SMILES. Furthermore, SELFIES is a string-based method, and so does not require as much cost as molecular graphs. A SELFIES-based molecular generative model, named GA+D, reportedly exhibited better performance than graph-based models in various benchmark tests, including single- and multi-objective optimisations for QED and penalised logP [22]. SELFIES has the potential to become a good solution for Ro5 outlier molecule generation. An existing weakness is the SELFIES collapse problem, in which different SELFIES strings are mapped to a single SMILES decoded from truncated SELFIES strings (Fig. A.1b). This collapse problem occurs because the SELFIES decoding algorithm ignores grammatically invalid symbols [23].

The SELFIES collapse problem can be a serious obstacle in molecular generation model training. Many goal-directed molecular generative models use single or multiple property scores, such as QED and logP. These property scoring functions generally require a SMILES string as an input [24]. Although a generative model produces SELFIES strings with a collapse problem, the generative model can receive positive feedback if the SMILES string corresponding to the generated SELFIES has good properties, resulting in unstable and unexpected training. Thus, to prevent this phenomenon in the molecular generative model design, the development of methods for alleviating the SELFIES collapse problem should be considered.

RL can be an effective tool in optimising molecular structures formatted as SELFIES because many string-based generative models, such as GENTRL [5] and ReLeaSE [4], achieved successful molecular optimisation using RL. Monte-Carlo policy gradient algorithm [25] is a RL method commonly used in molecular optimisation. In the algorithm, states and actions represent molecular structures and atom/bond modifying operations, respectively, and a reward is defined based on target property scores, such as logP and QED. Monte-Carlo policy gradient algorithm was widely used, but it suffers from the high variance problem of gradient, which can lead to unstable training and difficulty in finding optimal policy. To address this high variance problem, several extensions of Monte-Carlo policy gradient algorithm were developed, such as actor–critic [26] and self-critical sequence training (SCST) algorithm [27]. SCST is an extension specialised to effectively reduce variance in string-based RL tasks. It utilises the output of its greedy-based inference algorithm to normalise rewards for training, which alleviates the high variance problem and accelerates the model training.

In this study, we propose a SELFIES-based *de novo* drug design framework called ReBADD-SE to efficiently produce Ro5-complying drugs, but also molecules violating Ro5. Our methodological contributions include the novel SELFIES parsing algorithm for learning complicated SELFIES strings and the modified SCST algorithm for chemical sequence training. ReBADD-SE exploits a stack-augmented gated recurrent unit (stackGRU) network [28] with the proposed algorithms to generate SELFIES strings optimised with multiple objectives. Two benchmark datasets were used to evaluate ReBADD-SE. The first dataset contained GSK3b and JNK3 inhibitor data. The predictive models provided in [8] was used to evaluate the inhibitory activity of generated molecules to GSK3b and JNK3. The second dataset contained Bcl-2 family inhibitor data, including navitoclax and ABT-737 (Appendix B). To evaluate generated molecules, we obtained binding affinity experimental data from public databases [29–31] and exploited them to build a predictive model that can calculate the values of dissociation constant (pKd). Our results demonstrate that ReBADD-SE outperforms existing multi-objective molecular generation models in both benchmark tests. Furthermore, the SELFIES fragment parsing algorithm significantly mitigates the SELFIES collapse problem.

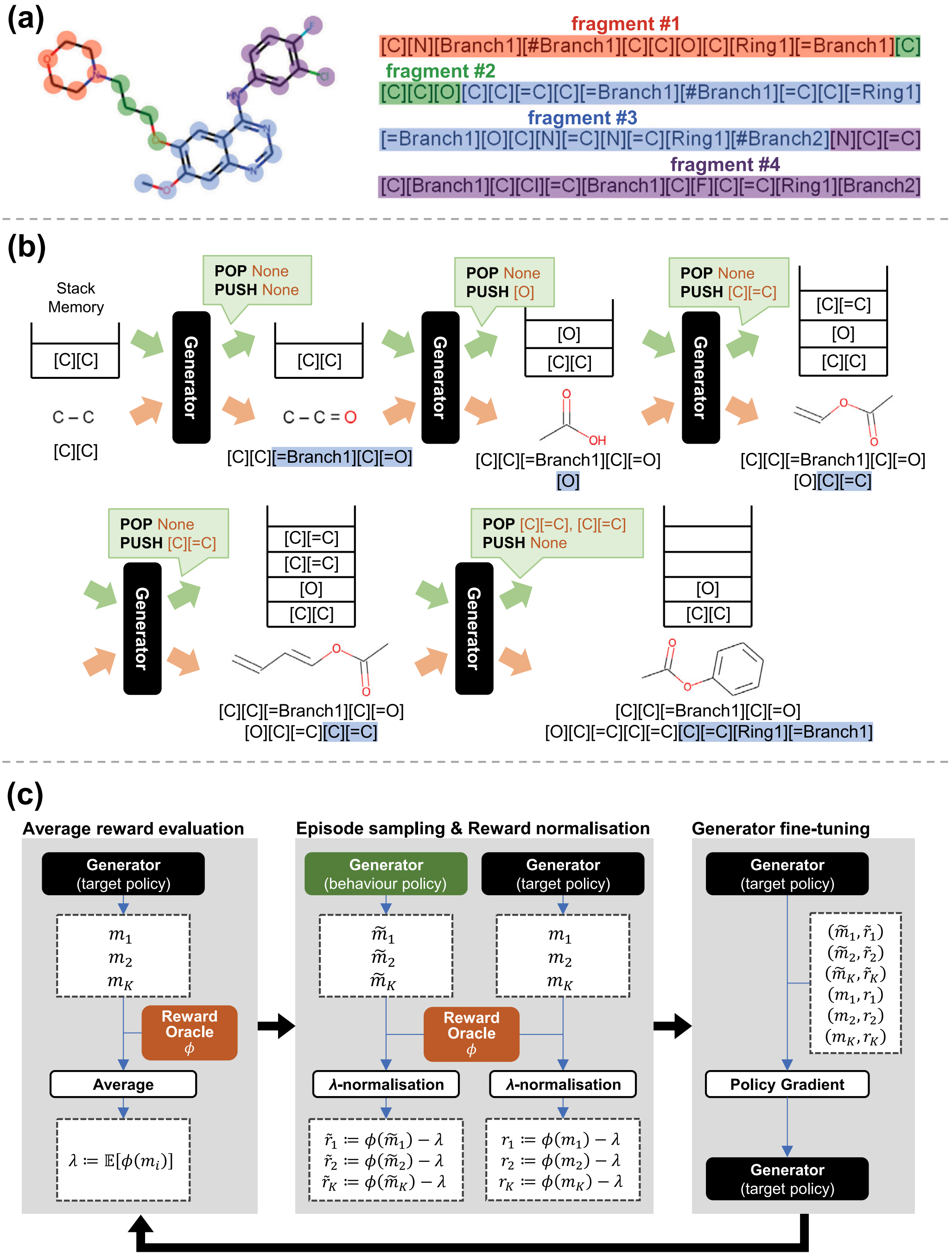
(a)
fragment #1
[C][N][Branch1][#Branch1][C][C][O][C][Ring1][=Branch1][C]
fragment #2
[C][C][O][C][C][=C][C][=Branch1][#Branch1][=C][C][=Ring1]
fragment #3
[=Branch1][O][C][N][=C][N][=C][Ring1][#Branch2][N][C][=C]
fragment #4
[C][Branch1][C][Cl][=C][Branch1][C][F][C][=C][Ring1][Branch2]
HN
N
N
O
Cl
F
(b)
Stack Memory
[C][C]
C – C
[C][C]
Generator
POP None
PUSH None
[C][C]
C – C = O
[C][C][=Branch1][C][=O]
Generator
POP None
PUSH [O]
[O]
[C][C]
OH
O
[C][C][=Branch1][C][=O]
[O]
Generator
POP None
PUSH [C][=C]
[C][=C]
[O]
[C][C]
O
O
[C][C][=Branch1][C][=O]
[O][C][=C]
POP None
PUSH [C][=C]
Generator
[C][=C]
[C][=C]
[O]
[C][C]
O
O
[C][C][=Branch1][C][=O]
[O][C][=C][C][=C]
POP [C][=C], [C][=C]
PUSH None
Generator
[O]
[C][C]
O
O
[C][C][=Branch1][C][=O]
[O][C][=C][C][=C][C][=C][Ring1][=Branch1]
(c)
Average reward evaluation
Generator (target policy)
m_1
m_2
m_K
Reward Oracle ϕ
Average
$\lambda \coloneqq \mathbb{E}[\phi(m_i)]$
Episode sampling & Reward normalisation
Generator (behaviour policy)
Generator (target policy)
$\tilde{m}_1$
$\tilde{m}_2$
$\tilde{m}_K$
m_1
m_2
m_K
Reward Oracle ϕ
λ-normalisation
λ-normalisation
$\tilde{r}_1 \coloneqq \phi(\tilde{m}_1) - \lambda$
$\tilde{r}_2 \coloneqq \phi(\tilde{m}_2) - \lambda$
$\tilde{r}_K \coloneqq \phi(\tilde{m}_K) - \lambda$
$r_1 \coloneqq \phi(m_1) - \lambda$
$r_2 \coloneqq \phi(m_2) - \lambda$
$r_K \coloneqq \phi(m_K) - \lambda$
Generator fine-tuning
Generator (target policy)
$(\tilde{m}_1, \tilde{r}_1)$
$(\tilde{m}_2, \tilde{r}_2)$
$(\tilde{m}_K, \tilde{r}_K)$
(m_1, r_1)
(m_2, r_2)
(m_K, r_K)
Policy Gradient
Generator (target policy)

Fig. 2. Overview of ReBADD-SE. (a) SELFIES fragment parsing based on SELFIES grammar rules. (b) Fragment-based SELFIES generation using stack-augmented GRU. (c) Off-policy self-critical sequence training for molecular optimisation.

2. Methods

The three main components of ReBADD-SE are (1) the SELFIES fragment parsing algorithm, (2) fragment-based SELFIES generator, and (3) molecular optimisation via the off-policy SCST algorithm (Fig. 2).

2.1. SELFIES fragment parsing

A SELFIES string consists of atom symbols (e.g. [C], [O], and [=C]) and structure symbols (e.g. [Ring1] and [Branch1]). While only one symbol is required to represent an atom in SELFIES, at least two symbols are required to represent the branch and ring structures. The first symbol in the sequence for structure representation determines whether the ring or branch and the next symbols are used to calculate the size of the structure, which is a crucial factor in the SELFIES collapse. More specifically, for a given SELFIES string $x = x_1 x_2 \cdots x_T$, if x_t is a structure symbol with N (e.g. [Branch$N$]), then the size of the structure beginning with x_t is computed by:

$$S(x_t) = 1 + \sum_{i=1}^{N} \mathbb{h}(x_{t+i}) \cdot |\mathbb{h}|^{N-i} \tag{1}$$

where $\mathbb{h}$ is the hash function mapping the SELFIES symbols to the corresponding integer values and $|\mathbb{h}|$ is the domain cardinality of the hash function. The mapping table for $\mathbb{h}$ can be found in the official SELFIES GitHub [21]. Using Eq. (1), we spliced a SELFIES string and parsed SELFIES fragments that were either complete branches or stem

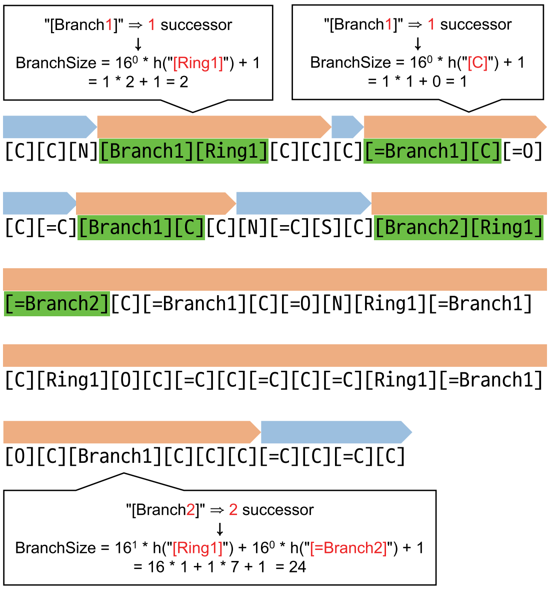
"[Branch1]" ⇒ 1 successor
↓
BranchSize = 16⁰ * h("[Ring1]") + 1
= 1 * 2 + 1 = 2
"[Branch1]" ⇒ 1 successor
↓
BranchSize = 16⁰ * h("[C]") + 1
= 1 * 1 + 0 = 1
[C][C][N][Branch1][Ring1][C][C][C][=Branch1][C][=O]
[C][=C][Branch1][C][C][N][=C][S][C][Branch2][Ring1]
[=Branch2][C][=Branch1][C][=O][N][Ring1][=Branch1]
[C][Ring1][O][C][=C][C][=C][C][=C][Ring1][=Branch1]
[O][C][Branch1][C][C][C][=C][C][=C][C]
"[Branch2]" ⇒ 2 successor
↓
BranchSize = 16¹ * h("[Ring1]") + 16⁰ * h("[=Branch2]") + 1
= 16 * 1 + 1 * 7 + 1 = 24

Fig. 3. Example of SELFIES fragment parsing algorithm.

structures (Fig. 3 and Algorithm 1). A new SELFIES string can be easily generated by reassembling parsed fragments. As each fragment has complete information about the branch structures, this fragment-level approach guarantees grammatical validity during SELFIES generation.

Algorithm 1 SELFIES fragment parsing

```
Require: A SELFIES string $(c_1, c_2, \cdots, c_T)$
Ensure: A sequence of SELFIES fragments $x$
  $x \leftarrow []$                                  ▷ Set an empty list
  $s \leftarrow \varepsilon$                         ▷ Set an empty string
  $t \leftarrow 1$
  while $t < T + 1$ do
      if $c_t$ is not a branch symbol then
          $s \leftarrow$ concatenate$(s, c_t)$       ▷ Extend a string $s$
      else
          $x \leftarrow$ concatenate$(x, [s])$       ▷ Append $s$ to $x$
          $s \leftarrow \varepsilon$
          $\zeta \leftarrow$ calculateBranchSize$(c_t)$   ▷ Eq. (1)
          for $j = 1, \cdots, \zeta$ do
              $s \leftarrow$ concatenate$(s, c_t)$
              $t \leftarrow t + 1$
          end for
          $x \leftarrow$ concatenate$(x, [s])$
          $s \leftarrow \epsilon$
      end if
      $t \leftarrow t + 1$
  end while
```

2.2. Fragment-based SELFIES generator

ReBADD-SE exploits a beta-variational autoencoder (β-VAE) architecture [32] where the encoder and decoder are a bidirectional GRU [33] and unidirectional stackGRU [28], respectively. A stackGRU has a memory layer that can store and provide relationship information between SELFIES fragments using push-and-pop operations [28]. For a SELFIES string $x = x_1 x_2 \cdots x_T$, the encoder calculates the corresponding latent vector using the following equation:

$$h_t = \text{GRU}(x_t, h_{t-1}) \quad \forall t = 1, \ldots, T, \tag{2}$$

where h_t is the tth hidden layer of GRU, and:

$$z(x) = \mu(h_T) + \epsilon \cdot \sigma(h_T), \tag{3}$$

where μ and σ are affine transformations used to calculate the mean and standard deviation vectors, and ϵ is random noise sampled from the standard Gaussian distribution. The decoder generates a SELFIES string from latent vector z by sequentially predicting the next tokens. Eq. (4) describes how to calculate the initial hidden layer of the decoder $\hat{h}_0$ from z:

$$\hat{h}_0 = \text{Tanh}(\text{Affine}(z)), \tag{4}$$

where Tanh$(\cdot)$ is a hyperbolic tangent activation function and Affine$(\cdot)$ is an affine transformation with trainable parameters. Next, for each $t = 1, \ldots, T-1$, the memory layer s_t is computed using Eqs. (5)–(8):

$$u_t = \text{Tanh}(\text{Affine}(x_t)), \tag{5}$$

$$(w_t^{push}, w_t^{noop}, w_t^{pop}) = \text{Softmax}(\text{Affine}(\hat{h}_{t-1})), \tag{6}$$

where Softmax$(\cdot)$ is a softmax function and $w_t^{push} + w_t^{noop} + w_t^{pop} = 1$. In Eq. (5), u_t is a new information vector, and in Eq. (6), three weights w_t^{push}, w_t^{noop}, and w_t^{pop} play roles in storing new information, retraining existing information, and removing old information, respectively, and:

$$s_t[k] = (w_t^{push}, w_t^{noop}, w_t^{pop}) \cdot q_t[k], \tag{7}$$

$$q_t[k] = \begin{cases} (u_t, s_{t-1}[k], s_{t-1}[k+1]) & \text{if } k \text{ is top,} \\ (s_{t-1}[k-1], s_{t-1}[k], \mathbf{0}) & \text{if } k \text{ is bottom,} \\ (s_{t-1}[k-1], s_{t-1}[k], s_{t-1}[k+1]) & \text{otherwise,} \end{cases} \tag{8}$$

where $q_t[k]$ is the kth element of vector q_t, $\cdot$ is the inner product operator, and $\mathbf{0}$ is a zero vector. After computing s_t, the hidden layer $\hat{h}_t$ and the probability vector y_{t+1} are calculated as follows:

$$\hat{h}_t = \text{GRU}([x_t, s_t]; h_{t-1}), \tag{9}$$

$$y_{t+1} = \text{Softmax}(\text{Affine}(\hat{h}_t)). \tag{10}$$

Both the encoder and decoder were pretrained with the evidence lower bound (ELBO) loss [34] to learn how to generate SELFIES strings. After pretraining, the decoder is fine-tuned via an off-policy SCST algorithm to produce SELFIES strings with desired molecular properties.

2.3. Training algorithm

The ReBADD-SE generator has two training phases: the pretraining phase to learn SELFIES grammars and the fine-tuning phase to achieve multi-objective molecular optimisation.

2.3.1. Pretraining phase

In the pretraining phase, the ELBO loss [34], RMSProp optimiser [35], and β scheduler [32] are used. The pretraining loss function is defined as follows:

$$L(x) = \sum_{t=1}^{T-1} \text{NLL}(y_{t+1}, x_{t+1}) + \beta \text{KL}(z(x)), \tag{11}$$

where NLL is the negative log-likelihood loss, KL is the Kullback–Leibler divergence loss, and $z(x)$ is the latent vector of x. In this study, β which controls the convergence speed of KL loss to mitigate KL vanishing [36], monotonically increases from 0 to 1 over iterations.

2.3.2. Fine-tuning phase

We designed a new off-policy SCST algorithm by applying an off-policy scheme to the SCST algorithm [27] to efficiently explore a large chemical space and improve the model performance, such as the diversity of generated optimal molecules. The SCST algorithm is a reward normalisation method that reduces the variance of the gradients computed by the Monte-Carlo policy gradient algorithm [25]. An original SCST uses an on-policy scheme and normalises the reward values for *stochastically* generated strings by subtracting a bias value, which is the reward of strings *deterministically* generated by the greedy algorithm. In our off-policy SCST algorithm, the reward generated by the behaviour policy network was normalised by subtracting the reward generated by the target policy network. More specifically, we first define mathematical terminology. $G(z)$ and $\tilde{G}(z)$ are the probability distributions of SELFIES strings given a latent vector z, which are computed by the target and behaviour policy networks, respectively. A behaviour network is periodically updated during the model training by cloning G. In RL, a SELFIES string is an episode, and the probability distributions are policies. ϕ is an oracle for calculating the rewards of SELFIES. Designing ϕ appropriate for a specific task is essential to achieve successful reinforcement learning [37]. We designed reward functions for benchmark tasks heuristically, and the information of reward functions was provided in Appendix C. The normalised reward $R(m; z)$ of a SELFIES string m generated from z is defined as:

$$R(x; z) = \phi(m) - \lambda(z), \tag{12}$$

where $\lambda(z) = \mathbb{E}_{y\sim G(z)}[\phi(y)]$. The bias value $\lambda(z)$ in Eq. (12) was computed using only the target network. The objective function for fine-tuning is defined as:

$$L(z) = \mathbb{E}_{m\sim G(z)}[R(m; z)]. \tag{13}$$

Using importance sampling with the behaviour $\tilde{G}$ and the gradient estimation formula, the gradient of Eq. (13) can be written as:

$$\begin{aligned}\nabla L(z) = \frac{1}{2}\mathbb{E}_{m\sim G(z)}\left[\nabla \log p_G(m; z) R(m; z)\right] \\ + \frac{1}{2}\mathbb{E}_{\tilde{m}\sim \tilde{G}(z)}\left[\nabla \log p_G(\tilde{m}; z)\frac{p_G(\tilde{m}; z)}{p_{\tilde{G}}(\tilde{m}; z)} R(\tilde{m}; z)\right].\end{aligned} \tag{14}$$

Our off-policy SCST uses episodes generated by both target and behaviour networks because the target network that is promptly updated during training can present more limited optimal episodes, which improves exploitation, whereas the behaviour network that is intermittently updated can provide diverse non-optimal episodes, which improves exploration. The pseudocode for ReBADD-SE training is described in Algorithm 2. We also exploited the entropy regularisation term in the target network training to improve its exploration by facilitating diverse episode generation of the target network [38].

Algorithm 2 Training algorithm of ReBADD-SE

```
Require: Initial trainable weights of generator $\Theta$, reward oracle $\phi$,
  SELFIES training dataset $D$, learning rate $\eta$, scheduler $\beta$, number of
  iterations $N$, batch size $K$, period of behaviour update $\tilde{N}$
Ensure: Trained weights $\Theta$
  for $n = 1, \cdots, N$ do                              ▷ Start of pretraining phase
      $L \leftarrow 0$
      for $k = 1, \cdots, K$ do
          $x \leftarrow$ Sampling($D$)
          $z \leftarrow$ Encoder($x, \Theta$)            ▷ Reparameterization trick
          $y \leftarrow$ Decoder($z, \Theta$)
          $L \leftarrow L + \mathrm{NLL}(y, x) + \beta(n)\mathrm{KL}(z)$      ▷ ELBO loss
      end for
      $\Delta\Theta \leftarrow$ RMSProp($L, \Theta$)
      $\Theta \leftarrow \Theta - \eta\Delta\Theta$
  end for                                                ▷ End of pretraining phase
  for $n = 1, \cdots, N$ do                              ▷ Start of fine-tuning phase
      $\Delta\Theta \leftarrow 0$
      if $n \equiv 1 \pmod{\tilde{N}}$ then
          $\tilde{\Theta} \leftarrow \Theta$                 ▷ Behaviour policy update
      end if
      for $k = 1, \cdots, K$ do
          $z \leftarrow$ SamplingGaussianDistribution()
          $\tilde{m} \leftarrow$ Decoder($z, \tilde{\Theta}$)
          $m \leftarrow$ Decoder($z, \Theta$)
          $\tilde{R} \leftarrow \phi(\tilde{m}) - \lambda(z)$                ▷ Eq. (12)
          $R \leftarrow \phi(m) - \lambda(z)$                  ▷ Eq. (12)
          $\Delta\Theta \leftarrow \Delta\Theta + \nabla \log p_\Theta(m) R$      ▷ Eq. (14)
          $\Delta\Theta \leftarrow \Delta\Theta + \nabla \log p_\Theta(\tilde{m}) \frac{p_\Theta(\tilde{m})}{p_{\tilde{\Theta}}(\tilde{m})} \tilde{R}$
      end for
      $\Theta \leftarrow \Theta - \eta\Delta\Theta$
  end for
```

2.4. Inference algorithm

ReBADD-SE generates an optimal molecular structure with desired properties by sampling latent vectors z from the standard Gaussian distribution. Specifically, ReBADD-SE randomly samples K latent vectors, generates a SELFIES string for each latent vector, evaluates them using a reward function, and selects the best one with the highest reward score. Algorithm 3 shows the optimised SELFIES generation process in the ReBADD-SE.

Algorithm 3 Inference algorithm of ReBADD-SE

```
Require: Trained weights of generator $\Theta$, reward oracle $\phi$, repetition
  time $K$
Ensure: A generated molecule $x_{best}$
  $x_{best} \leftarrow \varepsilon$                      ▷ $\varepsilon$ is an empty string
  $r_{best} \leftarrow 0$
  for $k = 1, \cdots, K$ do
      $z \leftarrow$ SamplingGaussianDistribution()
      $x \leftarrow$ Decoder($z, \Theta$)
      $r \leftarrow \phi(x)$
      if $r > r_{best}$ then
          $x_{best} \leftarrow x$
          $r_{best} \leftarrow r$
      end if
  end for
```

3. Results and discussion

3.1. Implementation details

ReBADD-SE was implemented using Python 3.8, and several open-source tools, including PyTorch 1.12.1 and RDKit 2021.03.4. RDKit, an open-source tool for cheminformatics, was used to evaluate QED and SA scores and to draw molecular structures. PyTorch, an open-source deep learning framework, was used to construct and train the neural networks of ReBADD-SE. All experiments were conducted on Ubuntu 18.04.6 LTS with 64 GB of memory and a GeForce RTX 3090. On this computational environment, the model training times including pretraining and fine-tuning phases took about 17 h.

3.2. Datasets

To evaluate the generative performances of ReBADD-SE, we used two benchmark datasets for the multi-objective molecular optimisation tasks.

3.2.1. GSK3b, JNK3, QED, SA

The first benchmark dataset was a public dataset presented by [8]. The goal of the dataset was to identify GSK3b and JNK3 dual inhibitor candidates with high QED and low SA scores. Each tuple of the dataset contained a SMILES string and two inhibitory activity scores for GSK3b and JNK3. Inhibitory activity scores were evaluated using the random forest model provided in [8,39]. The SA score is evaluated using the estimator provided in [40]. The scores for GSK3b and JNK3 ranged from 0 to 1, and the positive threshold was 0.5. The QED and SA values lie in a half-open interval $[0, 1)$ and closed interval $[1, 10]$, respectively. A molecule with high QED value is considered a good drug-like molecule. A high SA score indicates that the synthesis of a molecule is difficult. In a benchmark test, the successful generation of desired molecules was determined by the criteria of GSK3b$\geq$ 0.5, JNK3$\geq$ 0.5, QED$\geq$ 0.6, and SA$\leq$ 4.0.

3.2.2. Bcl-2, Bcl-xl, Bcl-w

The second benchmark dataset was designed in this study by curating molecule data from the ZINC15 database [41]. The goal was to identify drug candidates with strong binding affinity for Bcl-2 family proteins. Each tuple in this dataset contained a SMILES string and three binding affinity scores for Bcl-2, Bcl-xl, and Bcl-w. Binding affinity scores were evaluated using a deep neural network model trained on public datasets curated from the BindingDB [29], Davis [30], and GLASS [31] databases. The model, named ReBADD-DTA, predicts pKd values lying on $\mathbb{R}$ where a high pKd score represents a strong binding affinity to a target protein. Detailed information regarding the binding affinity prediction model is provided in Appendix D. The criteria for the benchmark test were defined as Bcl-2$\geq$9.069, Bcl-xl$\geq$8.283, and Bcl-w$\geq$6.999, which are the average affinity values of known Bcl-2 family inhibitors. The targeted Bcl-2 family proteins and the referenced inhibitors are described in Appendix B.

3.3. Evaluation metrics

We used three evaluation metrics proposed in [8]: the success rate (SR), novelty (Nov), and diversity (Div).

1. **SR**: proportion of molecules passing the given criteria of the given benchmark test among the generated molecules. A better model for multi-objective molecular optimisation should have a higher SR. More specifically, for a given set of generated molecules $\mathcal{G}$ and criteria ψ, SR is calculated as follows:

$$SR = \frac{|\mathcal{G}_\psi|}{|\mathcal{G}|}, \tag{15}$$

where $\mathcal{G}_\psi \subseteq \mathcal{G}$ is a set of molecules that pass the criteria ψ.

2. **Nov**: proportion of novel molecular structures among the passed molecules. A generative model with a high novelty score is expected to produce structurally different molecules from the existing compounds. More specifically, for a given set of reference molecules $\mathcal{H}$, Nov is defined as:

$$Nov = \frac{1}{|\mathcal{G}_\psi|} \sum_{x \in \mathcal{G}_\psi} \mathbb{I}\left[\max_{y \in \mathcal{H}} \operatorname{sim}(x, y) < 0.4\right] \tag{16}$$

where $\operatorname{sim}(x, y)$ is the Tanimoto coefficient over the Morgan fingerprints of the two molecules, x and y.

3. **Div**: average pairwise diversity scores for passed molecules. A high Div score indicates that the generative model can produce diverse molecules. Div is calculated as follows:

$$Div = 1 - \frac{1}{\binom{|\mathcal{G}_\psi|}{2}} \sum_{x \in \mathcal{G}_\psi} \sum_{y \in \mathcal{G}_\psi \setminus \{x\}} \operatorname{sim}(x, y) \tag{17}$$

We used the geometric mean (HMean) of SR, Nov, and Div to evaluate the overall performance of each model. All metrics were calculated using 5000 molecules sampled from each model.

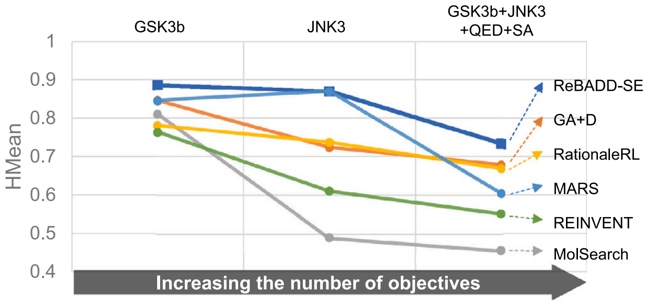

Fig. 4. Performance comparison over the number of objectives.

3.4. Model setup

Trainable parameters of ReBADD-SE were trained using Algorithm 2. Meta-parameters, such as the number of iterations, batch size, and learning rate, were heuristically determined (Appendix E). In the fine-tuning phase, we applied a linearly monotone increasing scheduler to the entropy regularisation terms to balance exploitation and exploration. We used a small step size for the behaviour policy update to guarantee that the target and behaviour policy distributions were not significantly different. Because the target and behaviour distributions become equal when the training loss converges, we simply implement Algorithm 2 by approximating $\frac{p_\Theta(\tilde{m})}{p_{\bar{\Theta}}(\tilde{m})}$ with 1, which increases the computational efficiency.

3.5. GSK3b, JNK3, QED, SA

We first evaluated ReBADD-SE using the public benchmark dataset for GSK3b and JNK3 dual inhibitor generation.

3.5.1. Experiment setup

We accessed the four metrics (SR, Nov, Div, and HMean) using two single-objective molecular optimisation tests and one multi-objective optimisation test. For each test, we trained ReBADD-SE, generated 5000 molecules 10 times, and compared the results to six baseline models, including GA+D [22], JTVAE [42], MolSearch [12], RationaleRL [8], MARS [9], and REINVENT [43].

3.5.2. Performance comparison

The results of GSK3b+JNK3+QED+SA tasks are presented in Table 1. The results of MolSearch and MARS were obtained by running their open-source codes, whereas the other baseline model results were obtained from published papers [8,9].

In single-objective optimisation tests, ReBADD-SE exhibited a nearly perfect SR of 95%–97% and demonstrated the best HMean scores of 0.89 ± 0.00 and 0.87 ± 0.00 for GSK3b and JNK3 inhibitor generation tasks, respectively. The scores of ReBADD-SE outperformed the HMean scores of baseline models with p-value < 0.01 evaluated by a one-sample t-test, excepting of the case of MARS in JNK3 single optimisation.

In the multi-objective optimisation task, ReBADD-SE still demonstrated the best HMean score of 0.73 ± 0.01, with a high SR of 84%. We confirmed that there were statistically significant differences between the HMean scores of ReBADD-SE and baselines with a p-value < 0.01, using a one-sample t-test. As shown in Fig. 4, the summary of the HMean results showed that even though the number of objectives was increasing, ReBADD-SE would be better than the baselines in multi-objective molecular optimisation.

Table 1
Performance comparison in GSK3b and JNK3 dual inhibitor generation. Results of MolSearch and MARS were obtained by running their open-source codes. Results of other baselines were taken from [9] and [8].

Model	Objectives	GSK3b				JNK3				GSK3b+JNK3+QED+SA			
		SR	Nov	Div	HMean	SR	Nov	Div	HMean	SR	Nov	Div	HMean
SELFIES	ReBADD-SE	0.97	1.00	0.71	**0.89**	0.95	0.91	0.76	**0.87**	0.84	0.69	0.67	**0.73**
	GA+D	0.85	1.00	0.71	0.85	0.53	0.98	0.73	0.72	0.86	1.00	0.36	0.68
Graph	JTVAE	0.32	0.12	0.90	0.33	0.24	0.03	0.88	0.18	0.05	1.00	0.28	0.25
	RationaleRL	1.00	0.53	0.89	0.78	1.00	0.46	0.86	0.74	0.75	0.57	0.70	0.67
	MARS	1.00	0.84	0.72	0.85	0.99	0.89	0.75	**0.87**	0.40	0.76	0.74	0.60
SMILES	REINVENT	0.99	0.61	0.73	0.76	0.99	0.32	0.73	0.61	0.48	0.56	0.62	0.55
	MolSearch	0.87	0.77	0.80	0.81	0.29	0.51	0.79	0.49	0.23	0.53	0.77	0.46

Table 2
Performance comparison in Bcl-2 family inhibitor generation.

Model	Objectives	Bcl-2+Bcl-xl+Bcl-w				Two sample
		SR	Nov	Div	HMean	t-test (*p*-value)
SELFIES	ReBADD-SE	0.9863 ± 0.0012	1.000 ± 0.000	0.515 ± 0.008	**0.798** ± **0.004**	–
Graph	MARS	0.1212 ± 0.0033	1.000 ± 0.000	0.908 ± 0.000	0.480 ± 0.004	177.8 (0.00)
	RationaleRL	0.3426 ± 0.0052	1.000 ± 0.000	0.831 ± 0.001	0.658 ± 0.003	88.5 (0.00)
SMILES	ReLeaSE	0.0002 ± 0.0002	0.700 ± 0.483	0.144 ± 0.244	0.018 ± 0.030	81.5 (0.00)
	MolGPT	0.0005 ± 0.0002	1.000 ± 0.000	0.608 ± 0.423	0.058 ± 0.041	56.8 (0.00)

Table 3
Ro5 violation analysis.

Model	MWT	LogP	Donor	Acceptor	Ro5-free score (↑)	t-test (*p*-value)
ReBADD-SE	941.16 ± 17.93	9.54 ± 0.23	2.84 ± 0.07	10.41 ± 0.21	**0.997** ± **0.003**	–
MARS	549.18 ± 0.16	4.60 ± 0.01	3.04 ± 0.01	8.61 ± 0.01	0.734 ± 0.002	230.7 (0.00)
RationaleRL	978.98 ± 1.21	10.95 ± 0.03	2.58 ± 0.02	9.77 ± 0.02	**0.997** ± **0.001**	0.0 (1.00)
ReLeaSE	527.04 ± 1.50	4.47 ± 0.02	1.16 ± 0.01	4.09 ± 0.02	0.292 ± 0.008	260.9 (0.00)
MolGPT	576.45 ± 1.09	5.33 ± 0.02	1.26 ± 0.02	6.38 ± 0.03	0.641 ± 0.008	131.8 (0.00)

Table 4
Binding affinity distribution analysis.

	Bcl-2		Bcl-xl		Bcl-w	
Model	Binding affinity	t-test (*p*-value)	Binding affinity	t-test (*p*-value)	Binding affinity	t-test (*p*-value)
ReBADD-SE	**9.41** ± **0.49**	-	**9.02** ± **0.33**	-	**8.23** ± **0.30**	–
MARS	8.27 ± 0.81	85.2 (0.00)	8.00 ± 0.73	90.0 (0.00)	7.31 ± 0.56	102.4 (0.00)
RationaleRL	8.77 ± 0.86	45.7 (0.00)	8.39 ± 0.73	55.6 (0.00)	7.23 ± 0.53	116.1 (0.00)
ReLeaSE	6.53 ± 0.84	209.4 (0.00)	6.58 ± 1.03	159.5 (0.00)	6.48 ± 0.76	151.4 (0.00)
MolGPT	6.56 ± 0.79	216.8 (0.00)	6.34 ± 0.86	205.7 (0.00)	6.03 ± 0.63	222.9 (0.00)

3.6. Bcl-2, Bcl-xl, Bcl-w

3.6.1. Experiment setup

Next, we evaluated the performance of ReBADD-SE in Bcl-2, Bcl-xl, and Bcl-w multi-target inhibitor generation tasks, which were more challenging than GSK3b and JNK3 dual inhibitor generation. In this study, we conducted a multi-objective molecular optimisation task because the candidates of Bcl-2 family inhibitors for senotherapy should be able to inhibit Bcl-2, Bcl-xl, and Bcl-w simultaneously [16–18].

3.6.2. Performance comparison

We selected MARS [9], RationaleRL [8], ReLeaSE [4], and MolGPT [44] as baselines and compared them with ReBADD-SE by generating 5000 molecules 10 times for each model. Table 2 presents the results. ReBADD-SE demonstrated the best performance, with an HMean of 0.80, while maintaining a SR of 99% and novelty of 100%, whereas all baselines had a SR < 50% and HMeans < 0.30. The gaps between ReBADD-SE and baselines were statistically significant with *p*-value < 0.01, which were evaluated by two-sample t-test. We found the reason why baselines had poor scores through two post-analysis experiments: Ro5 violation analysis (Table 3) and binding affinity distribution analysis (Table 4 and Appendix F).

3.6.3. Ro5 violation analysis

Based on the observation that many molecules with strong binding affinity against Bcl-2 family proteins violated more than one Lipinski's filter (Fig. A.1a), we defined a score called *Ro5-free score* that measures the proportion of molecules violating at least two Lipinski's filters among the generated molecules and evaluates the scores of ReBADD-SE and baselines. Since we obtained 10 sets of 5000 molecules for each model in the previous experiment, we reported the mean and standard deviation values in Table 3. ReBADD-SE and RationaleRL exhibited the highest Ro5-free scores of 0.997 ± 0.003 and 0.997 ± 0.001, respectively. ReLeaSE and MolGPT, which are SMILES-based generative models, showed the worst scores of 0.292 ± 0.008 and 0.641 ± 0.008, respectively. Excepting of RationaleRL, ReBADD-SE showed better Ro5-free scores than MARS, ReLeaSE, and MolGPT, with the significance of *p*-value < 0.01.

For a deeper analysis, we investigated the statistics of MWT, logP, number of hydrogen bond donors, and number of hydrogen bond acceptors over the generated molecules. ReBADD-SE and RationaleRL had extremely large values of MWT, LogP, and acceptors, which means the two models were able to produce Ro5-violating molecules easily. MARS tended to generate molecules violating the constraints on MWT and number of hydrogen acceptors, resulting in moderate Ro5-free score of 0.734 ± 0.002. ReLeaSE generated relatively heavy molecules with MWT > 500, but those generated molecules seemed to comply the other Lipinski's filters, resulting in a low SR of 0.02% (Table 2). As ReLeaSE was a SMILES-based recurrent neural networks, it might have difficulty in generating complicated molecular structures and thus had the worst Ro5-free score of 0.292 ± 0.008. MolGPT was also a SMILES-based generative model, but it exploited transformer networks and thus was able to produce molecules violating constraints on MWT, logP, or number of acceptors more effectively than ReLeaSE.

3.6.4. Binding affinity distribution analysis

Even though MARS, RationaleRL, and MolGPT had high Ro5-free scores (Table 3), they failed to achieve high SRs because they did not improve the objective properties sufficiently. For each model, we sampled 5000 molecules and computed the mean and standard deviation of their binding affinity values against Bcl-2, Bcl-xl, and Bcl-w (Table 4). ReBADD-SE outperformed all baselines by exhibiting 9.41 ± 0.49, 9.02 ± 0.33, and 8.23 ± 0.30 for Bcl-2, Bcl-xl, and Bcl-w, respectively. We used a two-sample t-test to evaluate the significance of gaps between ReBADD-SE and baselines and confirmed the *p*-value < 0.01 in all pairwise comparisons. Compared to the ZINC15 training data of 5.99 ± 0.67, 5.88 ± 0.79, and 5.70 ± 0.62 (Appendix F), MolGPT showed very small improvements of 6.56 ± 0.79, 6.34 ± 0.86, and 60.3 ± 0.63 for Bcl-2, Bcl-xl, and Bcl-w, respectively, resulting in a low SR of 0.05%. The two graph-based models, MARS and RationaleRL, demonstrated much better improvements in binding affinities than the SMILES-based baselines ReLeaSE and MolGPT, but MARS and RationaleRL had lower Bcl-2 scores of 8.27 and 8.77, respectively, than the threshold of 9.069 in the success criteria. As shown in Fig. F.1, we confirmed that many molecules generated by ReBADD-SE had stronger affinity scores than the existing Bcl-2 family inhibitors navitoclax and ABT-737.

3.6.5. Best candidates of Bcl-2 family of inhibitors

In drug discovery, it is important that the generated molecules have improved properties and synthetic feasibility. To evaluate the feasibility of ReBADD-SE molecules, we calculated the SA [40] and retrosynthetic accessibility (RA) scores [45] of each generated molecule; for each model, we selected the three best molecules with the highest total property (TP) scores (Table 5). The TP score was defined as the product of SA, RA, and the three binding affinity scores for Bcl-2 family proteins. Based on this definition, a molecule with strong binding affinity and good synthetic feasibility would have a high TP score. The top three molecules of ReBADD-SE had TP scores of 0.471, 0.447, and 0.441, respectively. The best score was significantly higher than all baseline outputs. The best molecules found by RationaleRL exhibited the strongest binding affinity values for Bcl-2. However, they had complicated molecular structures, resulting in poor SA (>0.4) and RA (<0.95) scores. Fig. 5 shows the top three molecules identified by ReBADD-SE with their TP scores.

3.7. SELFIES collapse analysis

ReBADD-SE mitigated the SELFIES collapse problem using SELFIES fragment parsing and fragment-level generation. To evaluate the degree of SELFIES collapse, we used Levenshtein distance to measure the difference between the two sequences (Fig. 6a). Specifically, the Levenshtein distance between two SELFIES strings is the minimum number of SELFIES character edits (insertions, deletions, and substitutions) required to change one string into another. The normalised Levenshtein distance is a distance value rescaled from 0 to 1. A severe collapse of a SELFIES string has a high normalised Levenshtein distance because many SELFIES characters are deleted when the SELFIES string is reconstructed from the corresponding SMILES string.

We sampled 5000 molecules for each model of the two SELFIES-based generative models ReBADD-SE and GA+D and compared their distributions of normalised Levenshtein distance values (Fig. 6b). The *degree of collapse* was defined as the mean of the normalised Levenshtein distance distribution. The degree of collapse for ReBADD-SE was 5%, which was significantly less than 88% for GA+D. Because the SELFIES fragments that ReBADD-SE used were grammatically parsed vocabularies from complete SELFIES strings, the strings produced by reassembling those fragments rarely violated SELFIES grammar. In contrast, GA+D produces SELFIES strings at the character-level without learning grammar, resulting in an extremely high collapse degree.

Table 5
Best property scores of feasible molecules generated by each model.

		Bcl-2	Bcl-xl	Bcl-w	TP
Model		SA	RA	–	Score†
ReBADD-SE	1st	9.17 3.06	9.12 0.96	8.44 –	**0.471**
	2nd	9.02 3.06	8.76 0.98	8.31 –	0.447
	3rd	9.18 3.31	8.81 0.99	8.28 –	0.441
RationaleRL	1st	10.17 3.97	9.65 0.95	7.96 –	0.450
	2nd	10.27 4.43	9.90 0.88	8.83 –	0.439
	3rd	10.12 4.19	9.44 0.94	7.68 –	0.401
MARS	1st	9.31 3.12	8.96 0.96	8.19 –	0.450
	2nd	8.94 3.66	9.08 0.97	8.34 –	0.415
	3rd	9.04 3.35	8.90 0.97	7.86 –	0.409
ReLeaSE	1st	9.52 3.24	8.62 0.98	7.91 –	0.431
	2nd	9.28 3.20	8.24 0.97	8.22 –	0.414
	3rd	8.35 2.46	8.27 1.00	7.68 –	0.399
MolGPT	1st	8.61 2.44	8.73 0.99	8.02 –	0.451
	2nd	9.25 2.65	9.12 0.94	7.67 –	0.449
	3rd	8.97 2.77	8.58 1.00	7.93 –	0.440

† TP = Bcl-2 * Bcl-xl * Bcl-w * (1 - 0.1 * SA) * RA

Those character-level generation cannot consider syntactic validation of SELFIES, such as branch representation, resulting in severe collapse problem (Fig. 6b). Whereas SELFIES generation with fragments parsed by our proposed algorithm lighten the burden of syntactic validation because our parsing algorithm presents only fragments complying the SELFEIS syntax, resulting in highly reduced collapse rate (Fig. 6b).

3.8. Ablation study

3.8.1. Char-level vs frag-level

The fragment-level SELFIES generation enabled faster training speeds compared to that of character-level generation. We trained two ReBADD-SE models on the Bcl-2 family dataset with and without the SELFIES fragment and evaluated their SR scores per checkpoint by sampling 1000 molecules from each checkpoint (Fig. 7). Both models achieved 95% SRs, but the fragment-level model converged more quickly. The fragment-level model demonstrated 97.3% SR at the 350-iteration checkpoint, whereas the character-level model showed 97.3% SR at the last 500-iteration checkpoint. The reason why character-level model training was slower was based on the differences in the

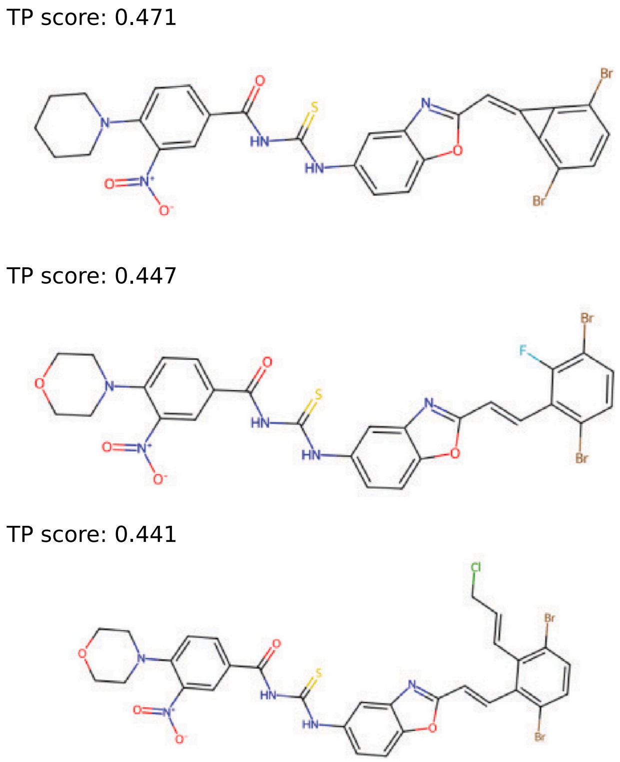

Fig. 5. Best three molecules generated by ReBADD-SE with good synthetic feasibility and strong binding affinity against Bcl-2 family proteins.

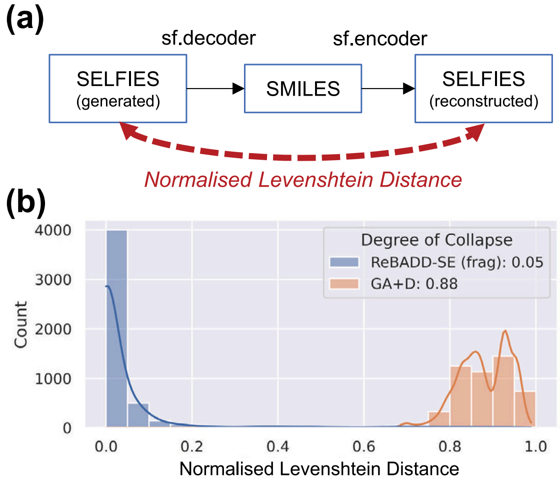

Fig. 6. SELFIES collapse analysis. (a) Description of normalised Levenshtein distance. (b) Histogram plots of normalised Levenshtein distance and degree of collapse.

information to learn. A character-level generation model should first explore valid molecular pieces, which can take a long time in the case of complicated and feasible structures such as Ro5 outliers. After exploring the valid pieces, the model resembled them to generate optimal molecular structures. In contrast, the fragment-level generative model could skip the exploring stage because those feasible pieces had already been parsed, and thus the model can focus on the generation stage, resulting in a fast training convergence speed.

3.8.2. On-policy vs off-policy

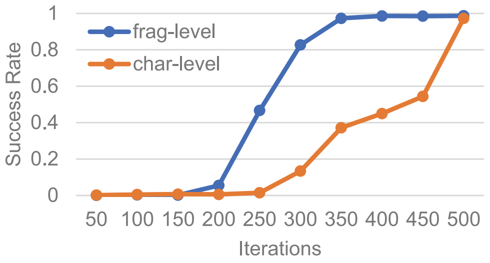

Fig. 7. Training convergence speed regarding of SELFIES tokenization.

The original SCST algorithm may malfunction in molecular optimisation tasks owing to the large size of the chemical space. The original SCST algorithm was designed with an on-policy scheme, which means that a single policy network plays a dual role in episode generation and policy gradient-based training. Because of the double role, if the model finds some desired episodes, it is apt to produce only the found episodes or similar ones, resulting in a limited SR and low diversity. In contrast, the off-policy SCST algorithm in ReBADD-SE utilises two policy networks. One is a behaviour policy that plays a role in episode generation. The other is a target policy that learns the episodes generated by the behaviour policy. Owing to the division of roles, even though the behaviour policy network finds the desired episodes, it still endeavours to find old and new optimal episodes without diversity degeneration because the behaviour policy does not promptly receive any feedback about the episodes.

To evaluate the two versions of ReBADD-SE trained by the original SCST and off-policy SCST algorithms, we used the Bcl-2 family dataset, sampled 1000 molecules per checkpoint, calculated their HMean scores, and visualised them (Fig. 8). In the visualisation, we used the extended connectivity fingerprints of diameter 6 (ECFP6), as suggested in a previous study [39] and embedded the fingerprints into two-dimensional space using a uniform manifold approximation and projection (UMAP) algorithm [46].

In original SCST, as the policy network immediately reflects the received feedback in episode generation, it is possible to quickly find episodes with improved rewards, but it is also fast to forget the found episodes due to the exploration on the immense chemical space, resulting in low generative performance (Fig. 8). In addition, as there are small number of molecules complying multiple property constraints, poor exploration performance due to the immediate feedback makes finding desirable episodes difficult. In contrast, ReBADD-SE with off-policy SCST showed slow improvement in the HMean score, but as the behaviour policy network remind the found desirable episodes and allow a target policy network to constantly learn the episodes, ReBADD-SE achieved a high HMean score of 0.804 and produced diverse molecular structures (Fig. 8).

4. Conclusion

This paper proposes a novel multi-objective molecular optimisation model, ReBADD-SE, to effectively produce Ro5-complying and Ro5-violating molecules. We exploited SELFIES representations to address complicated molecular structures and developed a fragment parsing algorithm to overcome the SELFIES collapse problem. Furthermore, we designed an off-policy SCST algorithm utilising a behaviour policy, verified its stable training performance, and demonstrated the superiority of ReBADD-SE via two benchmark tasks, GSK3b+JNK3+QED+SA and Bcl-2+Bcl-xl+Bcl-w.

Our significant methodological contributions include the novel SELFIES parsing algorithm and the improved SCST algorithm. We proposed those novel algorithms to address the SELFIES collapse problem and the limitation of original SCST. Both should be addressed to achieve effective multi-objective molecular optimisation. The combination of the proposed algorithms could resolve both problems simultaneously and exhibited its superiority in multi-objective molecular optimisation

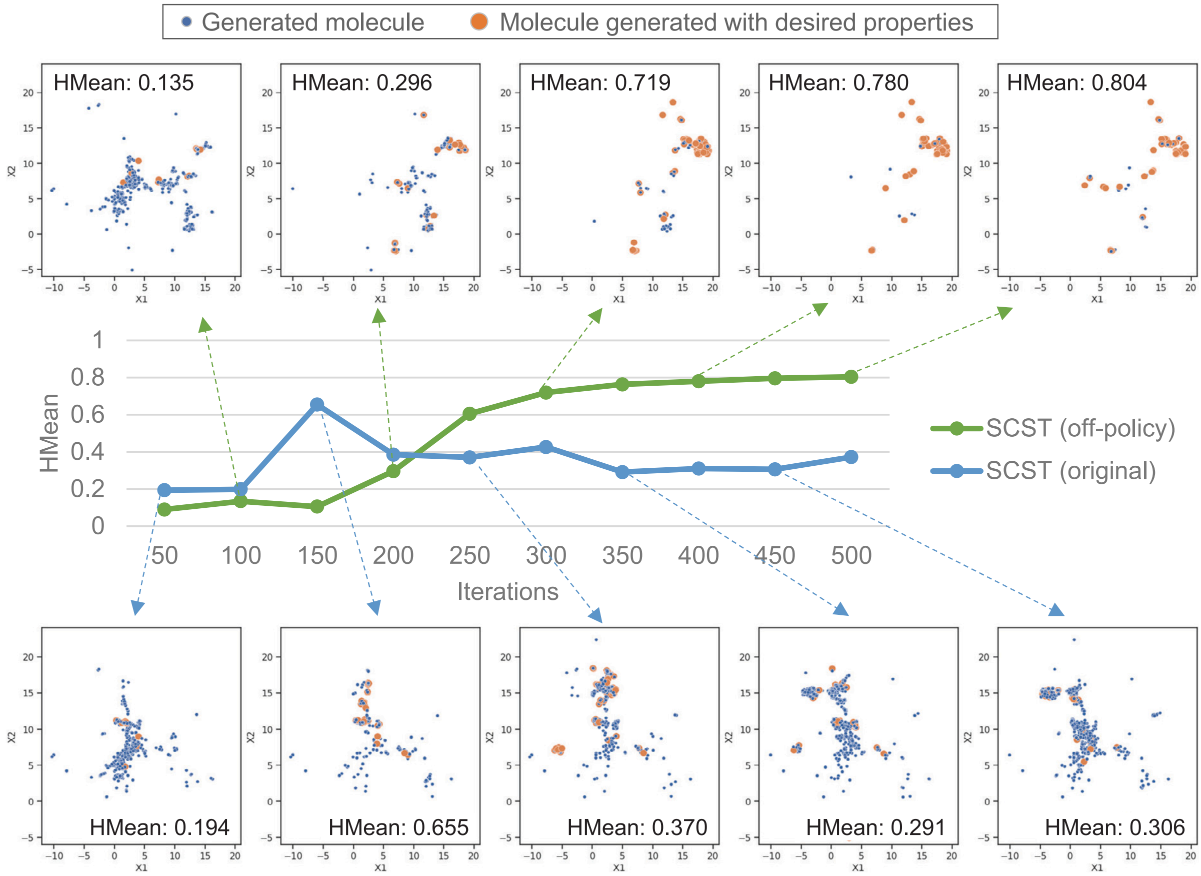

Fig. 8. Comparison between original SCST and off-policy SCST algorithms.

compared to the baseline methods (Tables 1–4). We expected that the proposed methods would present novel methodological ideas to peer researchers.

In this study, we showed that ReBADD-SE was effective to explore new hit molecules that can affect to multiple Bcl-2 family proteins. We considered only three binding affinity scores to find hit molecules, but more pharmacological properties, such as absorption, distribution, metabolism, excretion, and toxicity (ADME/T), should be considered together in order to discover drug-like molecules [47]. However, increasing the number of objectives makes molecular optimisation difficult because more sophisticated molecular generation techniques would be required and the number of available molecules satisfying all property constraints is small. Therefore, in the future work, we will make efforts to develop an improve multi-objective molecular optimisation method that can deal with large number of objectives simultaneously and find a method to train a generative model on a small size of dataset efficiently.

CRediT authorship contribution statement

Jonghwan Choi: Conception and design of study, Data preparation of benchmark datasets, Software implementation, Analysis of results, Drafting the manuscript. **Sangmin Seo:** Data preparation of benchmark datasets, Software implementation, Analysis of results. **Seungyeon Choi:** Software implementation. **Shengmin Piao:** Drafting the manuscript. **Chihyun Park:** Conception and design of study, Software implementation. **Sung Jin Ryu:** Conception and design of study, Analysis of results. **Byung Ju Kim:** Conception and design of study, Analysis of results. **Sanghyun Park:** Conception and design of study, Analysis of results, Reviewing manuscript.

Declaration of competing interest

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

Data availability

Our code for the experiments is available at https://github.com/mathcom/ReBADD-SE.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2019R1A2C3005212).

Appendix. Supplementary materials

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.compbiomed.2023.106721.

References

[1] P.G. Polishchuk, T.I. Madzhidov, A. Varnek, Estimation of the size of drug-like chemical space based on GDB-17 data, J. Comput. Aided Mol. Des. 27 (8) (2013) 675–679.

[2] K. Yamashita, M. Kaneko, M. Narukawa, A significant anticancer drug approval lag between Japan and the United States still exists for minor cancers, Clin. Pharmacol. Ther. 105 (1) (2019) 153–160.

[3] S. Romashkan, H. Chang, E.C. Hadley, National institute on aging workshop: repurposing drugs or dietary supplements for their senolytic or senomorphic effects: considerations for clinical trials, J. Gerontol. (A Biol. Sci. Med. Sci.) 76 (6) (2021) 1144–1152.

[4] M. Popova, O. Isayev, A. Tropsha, Deep reinforcement learning for de novo drug design, Sci. Adv. 4 (7) (2018) eaap7885.

[5] A. Zhavoronkov, Y.A. Ivanenkov, A. Aliper, M.S. Veselov, V.A. Aladinskiy, A.V. Aladinskaya, V.A. Terentiev, D.A. Polykovskiy, M.D. Kuznetsov, A. Asadulaev, et al., Deep learning enables rapid identification of potent DDR1 kinase inhibitors, Nature Biotechnol. 37 (9) (2019) 1038–1040.

[6] R. Winter, F. Montanari, A. Steffen, H. Briem, F. Noé, D.-A. Clevert, Efficient multi-objective molecular optimization in a continuous latent space, Chem. Sci. 10 (34) (2019) 8016–8024.

[7] J. Born, M. Manica, A. Oskooei, J. Cadow, M. Rodríguez Martínez, Paccmann rl: Designing anticancer drugs from transcriptomic data via reinforcement learning, in: International Conference on Research in Computational Molecular Biology, Springer, 2020, pp. 231–233.

[8] W. Jin, R. Barzilay, T. Jaakkola, Multi-objective molecule generation using interpretable substructures, in: International Conference on Machine Learning, PMLR, 2020, pp. 4849–4859.

[9] Y. Xie, C. Shi, H. Zhou, Y. Yang, W. Zhang, Y. Yu, L. Li, Mars: Markov molecular sampling for multi-objective drug discovery, 2021, arXiv preprint arXiv:2103.10432.

[10] T. Fu, C. Xiao, X. Li, L.M. Glass, J. Sun, Mimosa: Multi-constraint molecule sampling for molecule optimization, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, 2021, pp. 125–133.

[11] S.R. Krishnan, N. Bung, S.R. Vangala, R. Srinivasan, G. Bulusu, A. Roy, De novo structure-based drug design using deep learning, J. Chem. Inf. Model. (2021).

[12] M. Sun, J. Xing, H. Meng, H. Wang, B. Chen, J. Zhou, MolSearch: Search-based multi-objective molecular generation and property optimization, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022, pp. 4724–4732.

[13] B. Robson, J. Li, R. Dettinger, A. Peters, S.K. Boyer, Drug discovery using very large numbers of patents. General strategy with extensive use of match and edit operations, J. Comput. Aided Mol. Des. 25 (2011) 427–441.

[14] C.A. Lipinski, Lead-and drug-like compounds: the rule-of-five revolution, Drug Dis. Today: Technol. 1 (4) (2004) 337–341.

[15] C.A. Lipinski, Rule of five in 2015 and beyond: Target and ligand structural limitations, ligand chemistry structure and drug discovery project decisions, Adv. Drug Deliv. Rev. 101 (2016) 34–41.

[16] C. Tse, A.R. Shoemaker, J. Adickes, M.G. Anderson, J. Chen, S. Jin, E.F. Johnson, K.C. Marsh, M.J. Mitten, P. Nimmer, et al., ABT-263: a potent and orally bioavailable Bcl-2 family inhibitor, Cancer Res. 68 (9) (2008) 3421–3428.

[17] R. Yosef, N. Pilpel, R. Tokarsky-Amiel, A. Biran, Y. Ovadya, S. Cohen, E. Vadai, L. Dassa, E. Shahar, R. Condiotti, et al., Directed elimination of senescent cells by inhibition of BCL-W and BCL-XL, Nature Commun. 7 (1) (2016) 1–11.

[18] Y. Zhu, T. Tchkonia, H. Fuhrmann-Stroissnigg, H.M. Dai, Y.Y. Ling, M.B. Stout, T. Pirtskhalava, N. Giorgadze, K.O. Johnson, C.B. Giles, et al., Identification of a novel senolytic agent, navitoclax, targeting the bcl-2 family of anti-apoptotic factors, Aging Cell 15 (3) (2016) 428–435.

[19] Y. Ovadya, T. Landsberger, H. Leins, E. Vadai, H. Gal, A. Biran, R. Yosef, A. Sagiv, A. Agrawal, A. Shapira, et al., Impaired immune surveillance accelerates accumulation of senescent cells and aging, Nature Commun. 9 (1) (2018) 1–15.

[20] D. Flam-Shepherd, K. Zhu, A. Aspuru-Guzik, Language models can learn complex molecular distributions, Nature Commun. 13 (1) (2022) 1–10.

[21] M. Krenn, F. Häse, A. Nigam, P. Friederich, A. Aspuru-Guzik, Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation, Mach. Learn.: Sci. Technol. 1 (4) (2020) 045024.

[22] A. Nigam, P. Friederich, M. Krenn, A. Aspuru-Guzik, Augmenting genetic algorithms with deep neural networks for exploring the chemical space, in: ICLR 2020, 2020.

[23] W. Gao, T. Fu, J. Sun, C.W. Coley, Sample efficiency matters: Benchmarking molecular optimization, in: ICML 2022 2nd AI for Science Workshop, 2022, pp. 1–50.

[24] G. Landrum, et al., RDKit: A software suite for cheminformatics, computational chemistry, and predictive modeling, Greg Landrum 8 (2013).

[25] R.J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Reinf. Learn. (1992) 5–32.

[26] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT Press, 2018.

[27] S.J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, V. Goel, Self-critical sequence training for image captioning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7008–7024.

[28] A. Joulin, T. Mikolov, Inferring algorithmic patterns with stack-augmented recurrent nets, Adv. Neural Inf. Process. Syst. 28 (2015).

[29] M.K. Gilson, T. Liu, M. Baitaluk, G. Nicola, L. Hwang, J. Chong, BindingDB in 2015: a public database for medicinal chemistry, computational chemistry and systems pharmacology, Nucleic Acids Res. 44 (D1) (2016) D1045–D1053.

[30] M.I. Davis, J.P. Hunt, S. Herrgard, P. Ciceri, L.M. Wodicka, G. Pallares, M. Hocker, D.K. Treiber, P.P. Zarrinkar, Comprehensive analysis of kinase inhibitor selectivity, Nature Biotechnol. 29 (11) (2011) 1046–1051.

[31] W.K. Chan, H. Zhang, J. Yang, J.R. Brender, J. Hur, A. Özgür, Y. Zhang, GLASS: a comprehensive database for experimentally validated GPCR-ligand associations, Bioinformatics 31 (18) (2015) 3035–3042.

[32] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, A. Lerchner, Beta-VAE: Learning basic visual concepts with a constrained variational framework, in: International conference on learning representations, 2017, pp. 1–22, URL https://openreview.net/forum?id=Sy2fzU9gl.

[33] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014, arXiv preprint arXiv:1412.3555.

[34] D.P. Kingma, M. Welling, et al., An introduction to variational autoencoders, Found. Trends® Mach. Learn. 12 (4) (2019) 307–392.

[35] A. Graves, Generating sequences with recurrent neural networks, 2013, arXiv preprint arXiv:1308.0850.

[36] H. Fu, C. Li, X. Liu, J. Gao, A. Celikyilmaz, L. Carin, Cyclical annealing schedule: A simple approach to mitigating kl vanishing, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, pp. 240–250.

[37] J. Eschmann, Reward function design in reinforcement learning, Reinf. Learn. Algorithms: Anal. Appl. (2021) 25–33.

[38] Z. Ahmed, N. Le Roux, M. Norouzi, D. Schuurmans, Understanding the impact of entropy on policy optimization, in: International Conference on Machine Learning, PMLR, 2019, pp. 151–160.

[39] Y. Li, L. Zhang, Z. Liu, Multi-objective de novo drug design with conditional graph generative model, J. Cheminformatics 10 (1) (2018) 1–24.

[40] P. Ertl, A. Schuffenhauer, Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions, J. Cheminformatics 1 (1) (2009) 1–11.

[41] T. Sterling, J.J. Irwin, ZINC 15–ligand discovery for everyone, J. Chem. Inf. Model. 55 (11) (2015) 2324–2337.

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

[43] M. Olivecrona, T. Blaschke, O. Engkvist, H. Chen, Molecular de-novo design through deep reinforcement learning, J. Cheminformatics 9 (1) (2017) 1–14.

[44] V. Bagal, R. Aggarwal, P. Vinod, U.D. Priyakumar, Molgpt: Molecular generation using a transformer-decoder model, J. Chem. Inf. Model. 62 (9) (2021) 2064–2076.

[45] A. Thakkar, V. Chadimová, E.J. Bjerrum, O. Engkvist, J.-L. Reymond, Retrosynthetic accessibility score (RAscore)–rapid machine learned synthesizability classification from AI driven retrosynthetic planning, Chem. Sci. 12 (9) (2021) 3339–3349.

[46] L. McInnes, J. Healy, J. Melville, Umap: Uniform manifold approximation and projection for dimension reduction, 2018, arXiv preprint arXiv:1802.03426.

[47] L.L. Ferreira, A.D. Andricopulo, ADMET modeling approaches in drug discovery, Drug Dis. Today 24 (5) (2019) 1157–1165.