Magicmol - A light-weighted pipeline for drug-like molecule evolution and quick chemical space exploration

Lin Chen
Huzhou University
Qing Shen
Huzhou College
Jungang Lou (ljg@zjhu.edu.cn)
Huzhou University

Research Article

Keywords: Generative models, Reinforcement learning, Deep learning, Synthetic accessibility, De novo drug design

Posted Date: October 11th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-2124702/v1

RESEARCH

Magicmol - A light-weighted pipeline for drug-like molecule evolution and quick chemical space exploration

Lin Chen[1,3], Qing Shen[2,3] and Jungang Lou[1,3*]

*Correspondence: ljg@zjhu.edu.cn
[1]Zhejiang Province Key Laboratory of Smart Management & Application of Modern Agricultural Resources, School of Information Engineering , Huzhou University, Huzhou, China
[3]Yangtze Delta Region (Huzhou) Institute of Intelligent Transportation, Huzhou University, Huzhou, China
Full list of author information is available at the end of the article

Abstract

The flourishment of machine learning and deep learning methods have boosted the development of cheminformatics, especially when it comes to the application of drug discovery and new materials exploration. Lower time and space expenses make it possible for scientists to search the enormous chemical space. Recently, some work combines reinforcement learning strategies with an RNN-based (Recurrent Neural Networks) model to optimize the property of generated drug-like molecules, which notably improved a batch of critical factors for these drug candidates. However, a common problem of these RNN-based methods is that several generated molecules have difficulty in synthesizing even if owning higher desired properties such as binding affinity. But still, the RNN-based framework appears well in reproducing the molecule distribution among the training set than other categories of models when it comes to the molecule exploration tasks. Thus, to optimize the whole exploring process and make it contribute to the optimization of specified molecules. In this paper, we devised a light-weighted pipeline called - Magicmol with a re-mastered RNN network and use SELFIES presentation instead of SMILES. Our backbone model achieve extraordinary performance in evaluating metrics meanwhile reduced the training cost and we devised reward truncate strategies to eliminate the "model collapse" problem. Also, adopting SELFIES presentation makes it possible for combining STONED-SELFIES as a post-processing procedure for specified molecule optimization and quick chemical space exploration.

Keywords: Generative models; Reinforcement learning; Deep learning; Synthetic accessibility; De novo drug design

Introduction

The generative model is a class of techniques that use computational ways to devise molecules inversely, and these models can be roughly separated into several categories: variational autoencoders (VAEs) [1, 2, 3], generative adversarial networks (GANs) [4, 5], recurrent neural networks (RNNs) [6, 7], and flow-based models [8, 9]. In essence, the generative model learns valid molecule presentations from an extensive, cleaned database. For RNN-based models, while training, the input is like a "prefix" [10], for each time, a particular prefix is fed into the model, and the next character is defined as the training target. Concerning the existence of the hidden layer, which accounts for the ability of RNN models to process sequential data, RNN models take the output of the last step as the input of the next step and thus memorize the sequential information. During this process, an initial character

and a terminal character are used to indicate the start and termination of generating process. Following the established probabilistic rules, a well-trained generative model can "reproduce" the process while generating molecules with different sampling strategies (different sampling temperatures [11], etc.), which accounts for the validity and novelty of generated molecules.

Lots of experiments [12, 13, 14] had confirmed the feasibility and generative capacity of RNN-based models. And recently, Alan et all. conducted a research [15] , which made a comparison between chemical language models and got the conclusions that RNN-based models prevail over VAE-based models over reproducing the molecule distribution of the training set. However, that's not the whole story. The problem may cause by the biases of the training data thus leading the corresponding tasks to be grossly overestimated[16]; Without extra optimizations, RNN-based models appear well in common evaluate metrics, for instance, novelty, validity, and originality, but lots of generated molecules contain unwanted structures, or they are just not available for the reason of difficulty in synthesizing.

We see this problem is caused by the partiality of the evaluation metrics, especially the metric - novelty. Specifically speaking, the novelty of a generative model is defined as:

$$Novelty = 1 - \frac{|\text{set}\,(V_m) \cap N|}{|\text{set}\,(V_m)|} \tag{1}$$

where (V_m) is a batch of non-duplicate generated molecules, and N is the original training set. Any molecules that have never emerged in the training set are contributed to the novelty score. Concerning of formerly mentioned situation, such evaluating metric could not reflect this implicit structural problem, and it should be called the "permissive novelty" [17].

Meanwhile, Deep Generative Model (DGM) is not the only way for efficient chemical space exploration. Recently, some research utilized GA (Genomic Algorithm) such as Monte-Carlo Tree Search (MCTS) instead of DGM and proved GAs served as a potent candidate for searching for the desired chemical compounds[18, 19]. This search-based methods generally regard molecule fragments as the tree nodes, and the whole process can be viewed as searching for a feasible connection between the generated midbody and the nodes[20, 21]. The possibility for feasible connections not only ensure the validity of generated molecules but also make the search process efficient. However, the search process needs to get feedback from the third-part supervision, which could be a scoring function, a neural network[21], or some mechanism such as Expectation Maximization[22]. This step needs extra training and devise precisely for a reason to ensure it will lead the search process in the right direction.

In this paper, we tried a new pattern for combining the GA and DGM and treat novel molecule exploration as a two-step task. At first, we trained a 3-layer stacked RNN model as our backbone network for quick molecule exploration. Then we try to solve the problem of "permissive novelty" of RNN-based models using reinforcement learning while the Reinforcement Score (for details, see chapter Reinforce Score) is the target. And We adopt SELFIES [23] as the molecule presentation instead of

SMILES [24]. Second, when the network converged, we aim to utilize its exploration power to find the evolution target (a molecule with ideal properties), and the target will lead the optimization procedure for the specified molecule.

The main contribution of this paper are summarized below: (1) We re-trained a more efficient backbone model without inheriting the former framework and the post-processing bases on Reinforcement Score could make our model cater to different requirements. (2) We devised a light-weighted baseline that combines GA and DGM for specified molecule evolution without introducing extra parameters,the whole process is intervene-free and does not need further supervision. (3) We issued - reward truncate strategies for reducing the side-effect of reinforcement learning optimization and avoiding the model collapse and it could be transferred to other tasks.

Methods

Chemical molecular Presentation

For the purpose of making the model learns inner connection and presentation of valid chemical compounds. A certain chemical molecular must be presented as a meaningful vector [25]. As formerly mentioned, our baseline aims to combine with STONED-SELFIES[26] for molecule exploration, so we tried SELFIES - a novel and robust molecule presentation for molecule encoding. The appearance of SELFIES mitigated the problem of the random invalidity of SMILES and ensured the validity after structure modification. Our experiment shows excellent results while adopting SELFIES for encoding molecules as network inputs(Table 1). It shows superiority not only in the validity of generated molecular but also reduced the training cost.

Dataset and processing

A larger dataset will provide a more abundant combination of molecule fragments, which empowered DGM to search the enormous chemical space. For these concerns, we used the data collected from ChEMBL30 (https://www.ebi.ac.uk/chembl/) [27] which contains more than 2.2 million compounds and 1.92 million small molecules. We derived all of the small molecules for data preprocessing. After data-cleaning by rdkit (https://www.rdkit.org/)) [28], remove the salts, and stereochemical information and filter outs SMILES strings that are far from the center of chemical space, and convert all SMILES to canonical form. Half of the remained 1.72 million molecules are randomly selected as the training set, given the scope of our model structure.

After that, we take the SELFIES package to convert all selected drugs to SELFIES presentation and make up a chemical "corpus" of size c. It records all chemical substructure such as $[Branch1\_1]$ or $[C + expl]$ and three placeholder named $< start >$,$< pad >$,$< end >$ which means start generation, padding, and generation ended separately. While training, we first decompose the SMILES presentation of all molecular in our cleaned dataset into small fragments. Every fragment corresponds to one token in c, the process is shown in Figure 1. We noted that not all molecules could be encoded as SELFIES presentation for several reasons, such as violating the set rules. We exclude these molecules. Before we feed the data into the model, the molecule is split by the SELFIES package and encoded into meaningful vectors according to the chemical "corpus." Here we choose the length of the

longest encoded molecular (l) of each batch as the final vector length. A placeholder pads any molecular that does not match the final length in order to facilitate the training. Finally, $< start >$ and $< end >$ were added to the head and tail of the encoded vector separately.

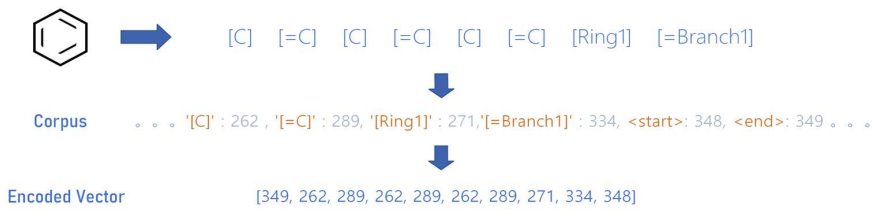


Figure 1 The encoding process and the example are given of a benzene molecule, it is first decomposed into items and structural tokens, and then each part is encoded by the established chemical corpus. Finally, the molecule is transferred to a meaningful vector.

Backbone model

A common problem emerged in DGMs is the invalidity of generated strings while adopting SMILES as the model input, and it is usually caused by the unmatched brackets[29]. The emergence of DeepSMILES[30] aims to solve this problem. And because of this, DGMs need more training epochs to reach the convergence and get rid of the invalidity problem. The former research such as ReLeaSE[31] deviced stack-memory layers to enhance the capacity of their model.

In our work, the modification of molecule presentation has changed and we defined our work as a light-weighted pipeline. Thus, we do not use the same generating networks with pre-trained weights, and we turn to train our backbone models and moved the former mentioned stack-memory layers; the workflow and model structure is shown in Figure 2. We adopt a 3-layer stacked recurrent neural network model with GRU [32]. GRU is another alternative solution with LSTM [33] which could ease the vanishing and explode of gradient [11] and thus make it possible to update more effectively during backpropagation. Compared with LSTM, it only contains two gates instead of three, thus reducing the training time and network parameters while not at the cost of model performance. The two gates of GRU are named "reset gate" and "update gate" separately, the reset gate means to control the information dependency of latest time h_t of last time h_{t-1}, while the update gate determines the extent information be reserved from last time h_{t-1}.

Here, we expect our model to learn the valid presentation while also being confined by the chemical properties such as chemical valance. Given a sequence of encoded vector $(V_1, ..., V_i)$, we let the model predict the distribution of the word $(V_i + 1)$. Take a common molecule as an example; if the model receives the sequence $c1ccccc$, we wish the model learns to maximize the probability distribution of the word 1, thus the ring is closed and yields the desired molecule. Formally speaking, given a vector V we try to maximize the probability of the equation

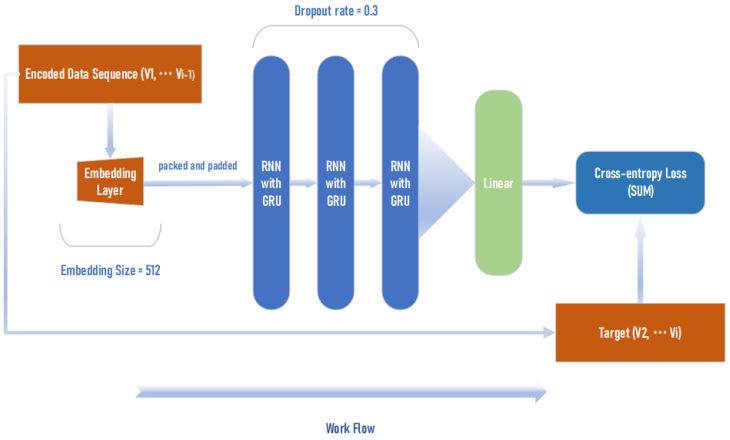


Figure 2 The data flow and backbone generative model for molecule generation

$$-\sum_{i=1}^{o} P(i) log_i P(V_{i+1}|V_1, V_2 ... V_i) \tag{2}$$

where P is the probability for each token in c be chosen as the next character and i is the time step.

After training, we sampled 10k molecules to evaluate the generative capacity of our model. We utilize principal component analysis (PCA) [34] and select the first two principal components and visualize them to confirm the exploring capacity of our generative model (Figure 3).

And the generation result illustrates that our model reached over 99% validity of generated molecules, diversity, and novelty, which are shown in Table 1 and we will discuss them in the next session.

Model Optimization

As we inferred, RNN-based models appear well in common evaluate metrics but lots of generated molecules contain unwanted structures. In Figure 4, we exhibited two molecules that may cause problems while conducting virtual screening [35] for the purpose of seeking proper drug-like chemical compounds. (the binding score is provided by IGEMDOCK [36], and the synthetic score is supplied by SYBA [37])

In silico molecule design can always be formulated as an optimization problem and it has been widely explorated[38, 39]. But the optimization may also be problematic. At first, multi-object optimization is still a problem in drug design field, because a certain compound must obey multiple physicochemical properties to be a drug candidate, and a single property being varied may lead to the changing of another property [7]. Second, pursuing too much on some properties may not work well, it's a bit comical that a molecule with the highest LogP would be such a long carbon string and of course is of no means for molecule design[40].

We seek the possibility of reducing the training cost while putting the model ahead and expect it generates molecules of high quality. Thus, we focus on optimizing only

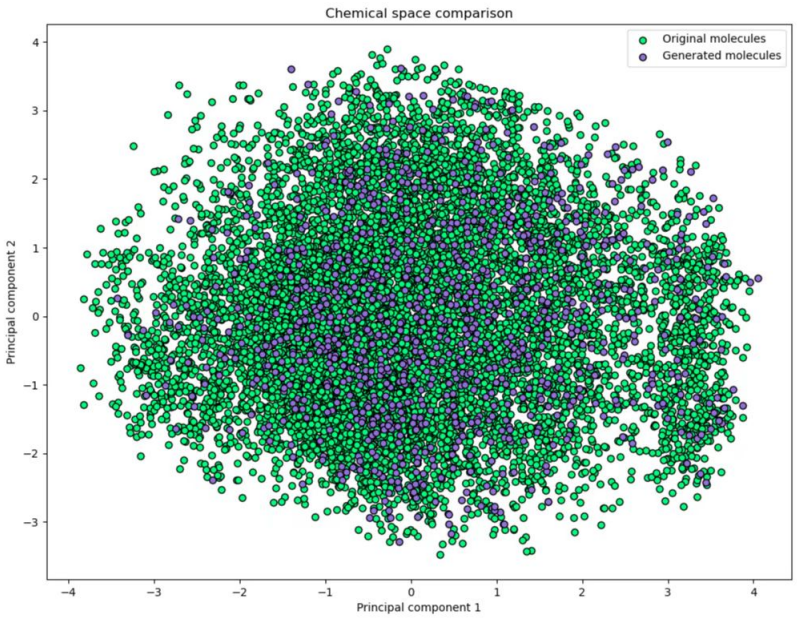


Figure 3 Visualization of training molecules and generated molecules among Chemical space using PCA, to make it more convenient for laying out, the data is 100 times diluted.

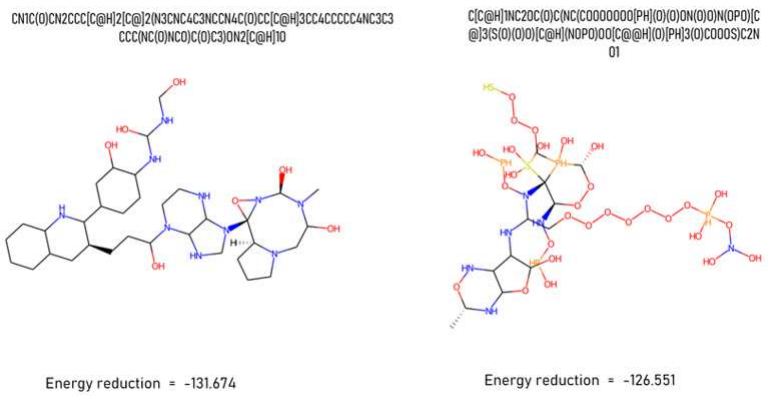


Figure 4 The deep learning model generates these molecules; both molecules seem to own higher binding capacity with the main protease of covid-19 (code:6LU7). The first molecule is difficult to synthesize because of its structural complexity, while the other has no meaningful chemical structure.

one important property - synthetic accessibility[41, 42], and the following reasons described our opinions: (1) we regard "the permissive novelty" as a problem caused by lacking a structural constraint, concerning the structure of drug-like molecules is often regular and easy to synthesize, from the economical side. Thus, the changing of SA (Synthetic Accessibility) may bring elevation of the quality of produced molecules. And for this purpose, we utilized SYBA instead of the traditional SA

score as our synthesis difficulty judgment. The design of SYBA takes the synthesis routes into concern and thus it could be a good quality evaluation tool for the generated molecules. (2) the next step of our pipeline may bring structure modification to a certain molecule, and this process may contain randomness. So these changes may deteriorate the structure of the variants thus we explicitly optimize it at first to mitigate this problem. (3) Treating SA as the optimizion object will bring our model interesting capacity and make it cater to different tasks (see Optimization of different tasks).

Synthetic score prediction

To directly optimize the synthetic accessibility from the generative model, we need to get feedback from the generated molecules. For this purpose, we take syba to judge all sampled molecules after one epoch is finished. Syba is capable for the classification of organic compounds as easy-to-synthesize (ES) or hard-to-synthesize (ES) According to syba, 0 serves as the threshold while estimating whether a molecule is difficult to synthesize or not. If the syba score is positive, the molecule is considered to be ES; otherwise, it is deemed to be HS [37].

For this work, we first generated numerous molecules from the original backbone model. We observed that approximately one-third of molecules should be estimated as hard to synthesize (Figure 6). And the next section, we tried to focus on two opposite directions - 1. Make the generative molecules harder to synthesize. 2. Make the generative molecules easier to synthesize. The whole process is shown in Figure 5.

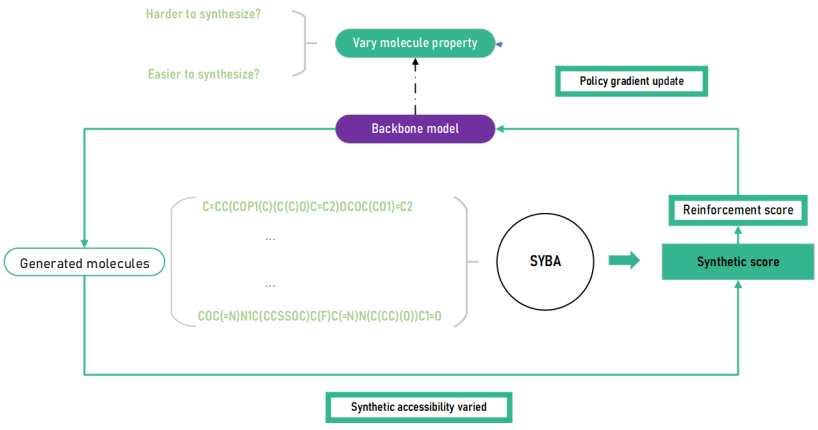


Figure 5 Workflow of property optimization. The generated molecules are judged by SYBA, their attributed score will be converted to a Reinforcement score using exponential projection. And then it will be utilized by the policy gradient optimization, with the molecule properties changing.

Optimization of different tasks

In our task, the reinforcement learning pipeline contains two modules: the actor and the critic. The actor takes current state (s_T) and performs an action (a_T) according to the environment, and the critic should provide feedback based on s_v and a_v thus conduct the actor be optimized in the right direction.

As for the traditional training process, the goal of the actor is to maximize the reward, and the reward equation is described in equation 3, and its derivative is listed in equation 4

$$R(\Theta) = \mathbb{E}\left[r\left(s_T\right)|_0, \Theta\right] = \sum_{s_T \in S} p_\Theta\left(s_T\right) r\left(s_T\right) \tag{3}$$

$$\nabla \bar{R}_\theta = \frac{1}{N} \sum_{n=1}^{N} \sum_{t=1}^{T_n} R\left(\tau^n\right) \nabla \log p_\theta\left(a_T^n \mid s_T^n\right) \tag{4}$$

$$\nabla \bar{R}_\theta = -\frac{1}{N} \sum_{n=1}^{N} \sum_{t=1}^{T_n} R\left(\tau^n\right) \nabla \log p_\theta\left(a_T^n \mid s_T^n\right) \tag{5}$$

and the model is trained to find a batch of parameters (Θ) to maximize the reward (R).

In our model, the current state s_t is acquired from each time step t according to the input token, and the action a_T is provided as the output of our backbone model. During this process, we sampled a group of action pairs (s_T,a_T) from which a brand-new molecule was de novo generated. In order to maximize the mathematical expectation $\mathbb{E}$, for each reinforcement training step, we generated 10 molecules. For each molecule, we accumulate the product of reinforcement score and action pairs so that we get the reward based on equation 3 mentioned above. Following the optimized rules, we force the model to "evolve" and modify its parameters, thus making the generated molecules own higher synthetic accessibility.

However, this process is not immutable, and the former works are always elaborated to maximize the mathematical expectation. In our work, we are delighted to observe that this process could also be reversed, which means with a slight adjustment, we can reduce the reward step by step and lead the model to vary in the opposite direction and reduce the synthetic accessibility. For this process, we follow equation 5 to conduct the adjustment. After training, we sampled 10k molecules from three models (backbone model(BM),negative-oriented optimized model(NM), and optimize-oriented model(OM)). And the generated result of all molecules is shown in Figure 6, and we can see a significant shift in the distribution.

Reinforcement score

According to the generated result of our original backbone model, the majority of synthetic scores of molecules distributes among the section from -150 to 150.

To facilitate the optimization process, an exponential projection is implemented to the original SYBA prediction, we call it the reinforcement score (RS) and use it to indicate a variant of synthetic accessibility for a certain molecule following a formula $e^{\frac{1}{150}x} + e$ where x refers to the predicted synthesis score predicted by SYBA. Based on the converting equation, a generated molecule with a higher predicted synthetic score will also have a higher reinforcement score. Before we take them into reinforcement optimization, we converted the synthetic score of all valid molecules into positive numbers.

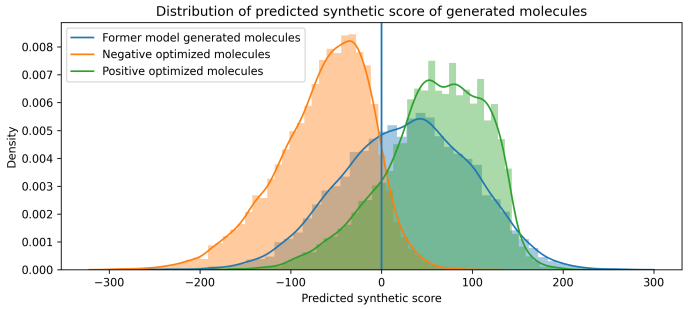


Figure 6 Prediction of the synthetic score by syba of three different stages: BM,NM,and OM. The blue part illustrates the scores derived from the original synthetic accessibility of molecules. There are 97.2% of molecules judged to belong to HS after negative optimization while 84.5% are believed to be ES after positive optimization.

Reward truncate strategies

Reinforcement learning can be viewed as a "post-processing" procedure for generative models, to be specific, the appointed reinforcement score indicates the model to change toward our desired direction. However, the procedure is delicate and difficult to control, and a phenomenon called "model collapse" affected the quality of generation, which often reflexes in too many duplicate tokens of generated molecules and a performance recession. We view this phenomenon as caused by the "stable revenue" of positive examples, the model could repeat such series and thus get a higher reward easily, and this problem is implicit and obscure.

For our task, some details deserve further discussion.

(1)We noticed that there are still parts of molecules "born" with high synthetic accessibility, from the optimal perspective, these molecules could waive further optimization.

(2)We also noticed that our appointed reinforcement score may be too smooth for differentiating HS molecules or ES molecules. As we former mentioned, SYBA regards 0 to be the boundary of two categories. The RS projected all SYBA predictions to other continuous spaces and after that all attributed RS is positive. But the modified continuous space becomes not obvious for differentiating the two categories. For example, molecules with -10 as a predicted SA score, and after the exponential projection, its RS will be $e^{-\frac{1}{15}} + e$; and a molecule with 10 as the former, its RS will because $e^{\frac{1}{15}} + e$. We can see that the difference seems too slight after the projection and we expect to provide the model a more clear instruction when it conducts the task of positive optimization.

To solve all mentioned problems, a truncation of the reinforcement score is utilized to ensure better training results. We try to utilize the ideology of activating functions such as Relu, which exerts a non-linear transformation of the given expression. Here we first set an "optimal threshold" to exclude these molecules while conducting action-pair sampling. Any ES molecules over this threshold will not contribute to the calculation of the next step. And in our experiment, this threshold is set to 150 (before converting it to the reinforcement score). For other ES examples, only half of their reinforcement score contributes to the calculation of the next step. Actually, we try to reduce the reward of these examples to some extent thus getting

rid of too many generated duplicate tokens. To evaluate this, we did experiment with differences on adding these strategies or not, and the comparison is shown in Figure 7. The result shows the issued strategies mitigated the phenomenon of 'model collapse' and ensured the richness of generated molecules.

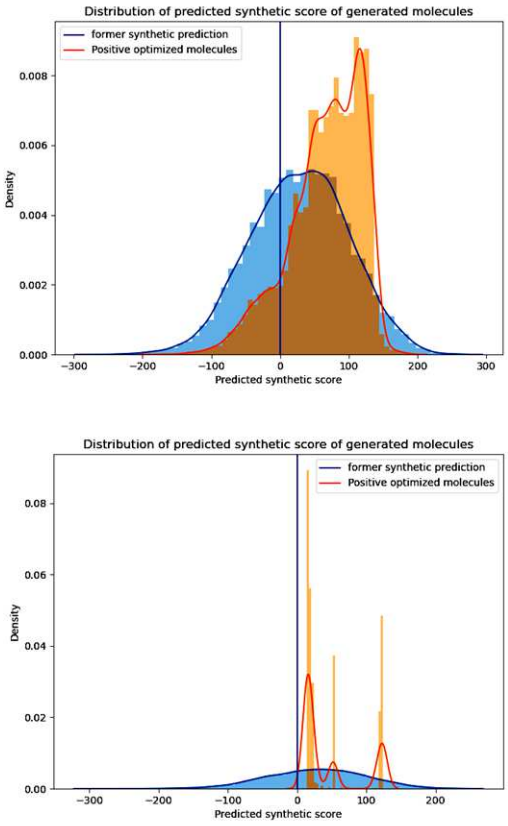


Figure 7 Comparison between model implemented our strategies and not after training 10 epochs. The post-processing is conducted on the same model with only utilizing the defined rules or not as the difference. For each model, we sampled 10 molecules at a single optimization timestep and repeat 20 times for a single epoch and the total epoch is set to 10 in our experiment. The upper figure shows that with proper strategies, the navigation of chemical space is feasible in comparison with the collapse model without any constraint.

Halfway-targeted drug-like molecules exploration

In the next part, we utilized STONED-SELFIES - an algorithm using structural evolution to quickly explore the medium molecules; Following the metrics issued in their works, the joint similarity ensures the evolution of the midbody molecule has similarities to their parents (for details, see the original paper [26]). And for our model, the light-weighted design empowers it to conduct quick exploration. Thus, we can use it to search the enormous chemical space efficiently and find the ideal molecule as the evolution endpoint. So the two parts can be combined together for designated molecule evolution. And the detail of our experiments are listed below:

(1) Following the former parts to train the generative model with positive optimization, then we conduct molecule selection to choose the best molecules with

ideal LogP and QED score (calculated by Rdkit). (2) Assign the "best" candidates as the evolution endpoint and drug candidates as the starting point to conduct structure evolution using STONED-SELFIES, see Figure8.

Thus, the pinpoint of our method lies in finding the ideal evolution target and the start point and the purpose is to do the structural evolution from one to another. During this process, a bunch of midbodies will be explored in the near chemical space. With proper selection strategies, we could find molecules with both ideal properties and similarities to their parents. And we called this method - Halfway-targeted drug-like molecule exploration.

Results

Speed and performance

In our work, we retrained a backbone network without using the same parameter or structure of former excellent works[43, 7, 31] and adopted the SELFIES presentation. Our model appears to have high performance throughout the whole training process with approximately 4.9 million parameters, only one-tenth of the backbone models of others[43]. Among our concepts, the sampling capacity matters[44], for the reason of efficiently searching for a proper evolution target, thus we take the model structure into consideration. The reduction of structures lets our model works rapidly without losing performance, even after the reinforcement optimization. While the model is working on a laptop with a graphic card (GTX 1060 with 6GB video memory), it still reached a high speed of generating approximately 1k novel molecules in less than a single second.

After training toward different expectations, we sampled 10k molecules from each model for 3 rounds and calculated some standard metrics in an average; the result is shown in Table 1.

The definitions of metrics are :

$$Uniqueness = \frac{|\text{set}\,(V_m)|}{|V_m|} \tag{6}$$

$$Validity = \frac{|V_m|}{S_m} \tag{7}$$

where S_m means a batch of sampled molecules and V_m represents chemical valid molecules for a single batch in S_m.

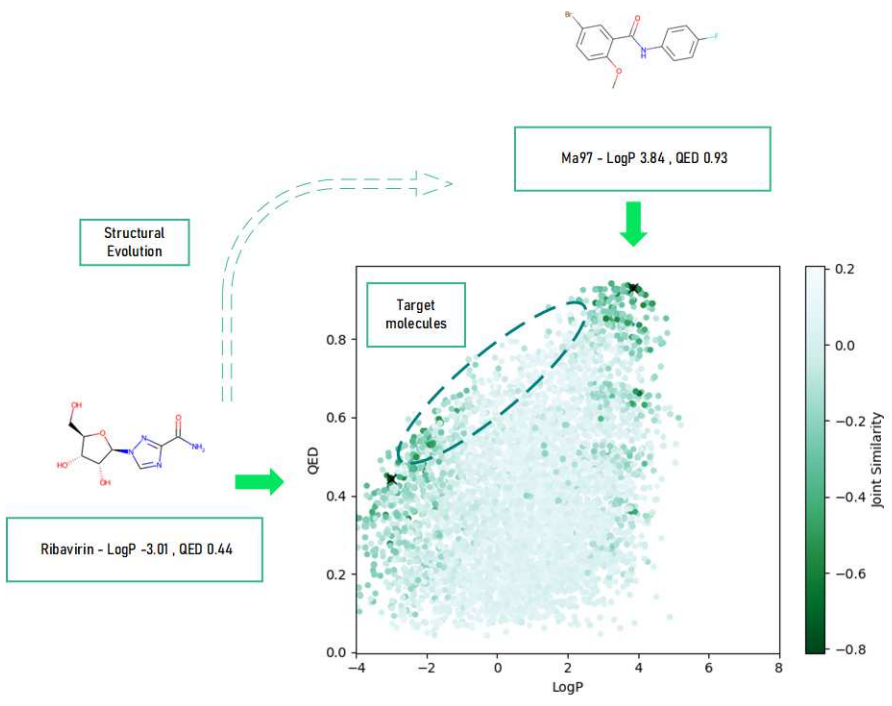
Table 1 Performance of Magicmol in different metrics. Synthesizability: generated molecules judged as ES by SYBA; Druglikeness: average QED score of generated molecules done by Rdkit

Models	Validity	Uniqueness	Novelty	Traning Set	Synthesizability	Druglikeness	Model Parameters
Magicmol (BM)	99.90%	99.90%	100%		0.655	0.14	
Magicmol (NM)	96.20%	96.20%	100%	0.89 M	0.027	0.51	4.9 M
Magicmol (OM)	99.90%	99.90%	100%		0.885	0.69	

Molecule Evolution

An example of a molecule evolution process is shown in Figure 8. The evolution starts with a drug candidate - Ribavirin (LogP -3.01 , QED 0.44) and ends with the generated molecule Ma97 (SMILES: COC1=CC=C(Br)C=C1C(=O)NC2=CC=C(F)C=C2 , LogP 3.84 , QED 0.93). During this process, STONED-SELFIES applied reasonable string manipulation. The modification of molecule presentation can be seen as a process of exploring the near chemical space around the specified molecule. And our designated molecule with ideal drug-like properties will lead the modified direction (the joint similarity plays as a structural constraint). Though the evolution process indeed has a randomness to some extent, but the advantage is also distinct. First, it's time-consuming and needs fewer computational resources, thus it can be replicated round by round to extensively explore the surrounding chemical space. Second, the evolution is structural-dense because at each timestep we only permit up to 2 tokens to do alternation thus the evolution is explainable and changes between different fragments can be detected and analyzed. Third, the whole process is done step-by-step, we first explicitly optimize the SA of generated molecules and then turn to optimize other chemical properties, we implicitly reach the goal of multi-object optimization which is also a dilemma in this field.

Figure 8 An evolution process between the formerly mentioned molecules. The result confirmed our ideas and we circled the molecules with better physicochemical properties in green dotted ellipse.



Discussions

In this paper, we proposed Magicmol, which focuses on utilizing the advantage of methods from two categories (the exploration capacity of DGMS and the evolution

abilities of GAs). We first designed an RNN-based backbone model and conducted optimization thus empowering generate molecules with ideal chemical structures, and then we combined them with STONED-SELFIES to do the molecule evolution to explore near chemical space to optimize a certain molecule. In the next section, we will discuss other opinions and potential applications of Magicmol.

De novo drug design

To be honest, Magicmol is not born for de novo devise drug-like molecules, and a potent drug candidate is a combination of several aspects such as logP, QED(quantitative estimate of drug-likeness), ADMET(Absorption, Distribution, Metabolism, Excretion, Toxicity), etc. We expect ideal drug-like molecules to own all these factors. As for the reinforcement learning optimization, a single property being varied may lead to the changing of another property [7], their solution is modifying these properties one by one, which works well but also takes higher time and computational complexity.

Thus, we tried to focus on altering only one significant property - synthetic accessibility, and we have conducted a series of experiments to assess the molecule properties after reinforcement learning optimization. Actually, there is no absolute evidence proving that the structural complexity is binding with the drug-like factors. Still, we witnessed a massive difference between these generated molecules after changing their synthetic accessibility to different directions Figure 9.

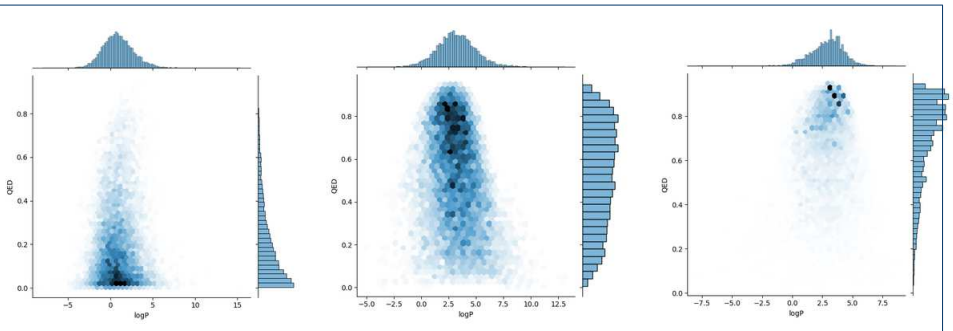


Figure 9 QED and logP score of generated molecules after reinforcement learning optimization. We derived 10 batches (10240) of molecules from our generative model : (a) NM (b) BM (c) OM ,and their QED and logP score are acquired by rdkit. There is a massive difference in some drug-like properties between these generated molecules, even if we just modified the synthetic accessibility.

Even though these properties are not perfectly positively correlated to synthetic accessibility, we still observed that accompanying with the structural optimization, the drug-likeness of generated molecules rises, especially for the averaged QED score, which increases from 0.51 (BM) to 0.69(OM). And this result matches our expectation; we see this elevating on some metrics is achieved by the "get feedback and optimize" pipeline, which modified the network parameters, thus enhancing the model capable of generating ideal molecules. Also, we think Magicmol mitigated the difficulty of retro-synthesize route identification to some extent concerning the less complex of these generative molecules.

Synthetic accessilbility variation

We tried to reverse the rule of policy gradient so that our model can be used to vary the synthetic accessibility in a different direction, which can be utilized for either improving the synthetic accessibility of molecules or simply generating lots of hard-to-synthesis compounds without introducing any other super parameters and we get rid of the need of domain knowledge. To the best of our knowledge, other models have not emerged that try to vary the property of molecules directly from the generative model to the opposite direction.

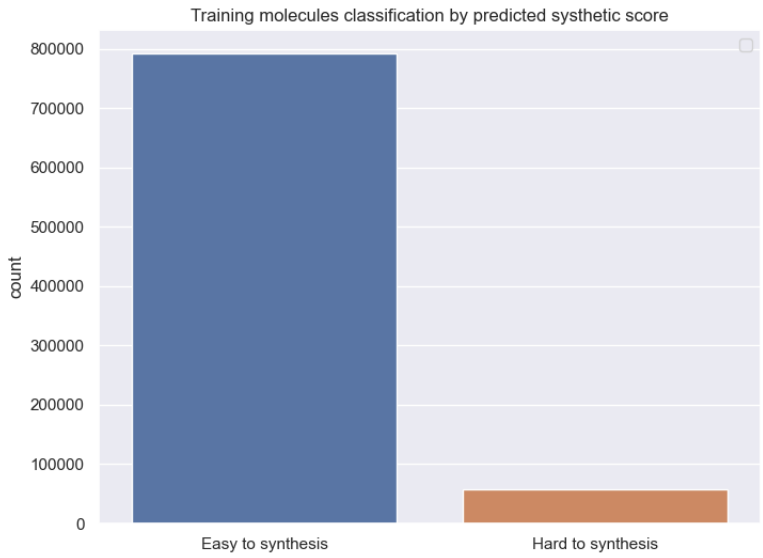


Figure 10 Distribution of training molecules by their predicted synthetic score, the positive samples are approximately 13 times the negative samples.

In the natural world, intuitively, most compounds are designed to be easier to synthesize. Then we tried to derive the synthetic score of the original training set, and the result is shown in Figure 10. And only approximately 7% of the compounds are judged to be difficult to synthesize. In some particular tasks, for example, the training of SYBA. For better training results, a model should reach a balance between negative samples and positive samples while preparing the training set. Magicmol may serve as a high-velocity negative sample generation tool, which could be a solution for such problems. And in Figure 11, we listed some generated molecules after the reinforcement optimization in different directions.

Conclusions

In this paper, we proposed Magicmol, which focuses on utilizing the advantage of methods from two categories (the exploring capacity of DGMS and the evolution abilities of GAs). The idea initially seems contradicted but actually can be reasonably combined. We empowered the model to generate molecules with ideal chemical structures but also utilized structural constraints that facilitate the following evolution steps. The pipeline could conduct quick exploration of enormous chemical space

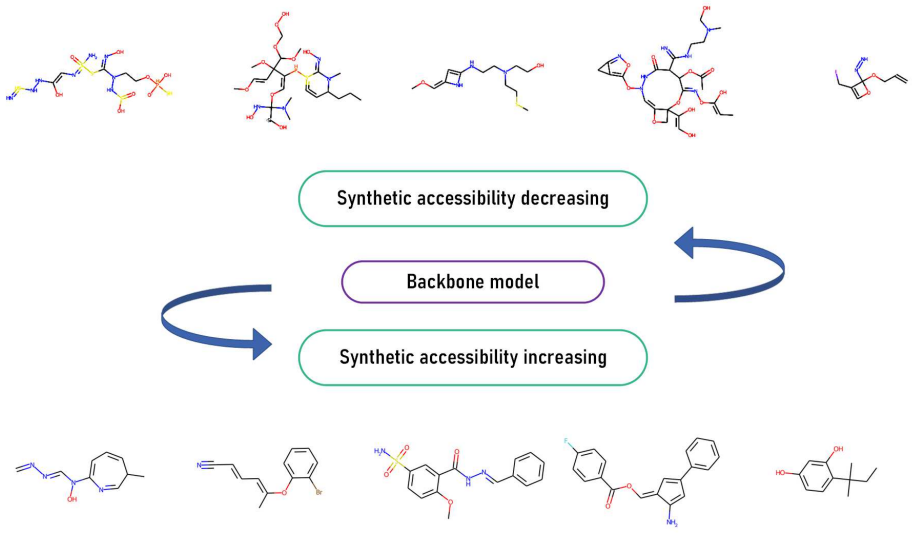


Figure 11 Generated molecules of different synthetic accessibility. The first line : the top 5 molecules generated with the highest synthetic accessilbility The second line: the top 5 molecules generated with the lowest synthetic accessilbility

and we also issued our solutions to solve the annoying 'model collapse' problem. Thus, we think Magicmol may solve problems in this field to some extent.

Abbreviations

BM:Backbone model;OM:Optimize-oriented model;NM:Negative-optimize oriented model; DGM:Deep Generative Model;GA: Genetic Algorithm;Reinforcement Score: SS; RNN:recurrent neural network; VAE:variational autoencoder; GAN:enerative adversarial network; syba: Synthetic Bayesian Classifier; GRU:gated recurrent unit;LSTM:Long Short-Term Memory; ES:easy-to-synthesize; HS:hard-to-synthesize; QED:quantitative estimate of drug-likeness; ADMET:Absorption,Distribution,Metabolism,Excretion,Toxicity

Acknowledgement

Not applicable.

Funding

This work was supported in part by Natural Science Foundation of Zhejiang Province under Grant LR20F020002, in part by the Primary Research and Development Plan of Zhejiang Province,Grant 2020C01097.

Availability of data and materials

The source codes of this article is available from https://github.com/Josefjosda/Magicmol repository. And training data can be derived from https://www.ebi.ac.uk/chembl/.

Declaration

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

Lin Chen initially issued the idea and he is responsible of the code application and paper writing. Jungang Lou and Qing Shen supervised this study and prepared the manuscript. Both authors read and approved the final manuscript.

Author details
[1]Zhejiang Province Key Laboratory of Smart Management & Application of Modern Agricultural Resources, School of Information Engineering , Huzhou University, Huzhou, China. [2]School of Science and Engineering, Huzhou College, Huzhou, China. [3]Yangtze Delta Region (Huzhou) Institute of Intelligent Transportation, Huzhou University, Huzhou, China.

References

1. Gómez-Bombarelli, R., Wei, J.N., Duvenaud, D., Hernández-Lobato, J.M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T.D., Adams, R.P., Aspuru-Guzik, A.: Automatic chemical design using a data-driven continuous representation of molecules. ACS central science **4**(2), 268–276 (2018)
2. Lee, M., Min, K.: Mgcvae: Multi-objective inverse design via molecular graph conditional variational autoencoder. Journal of Chemical Information and Modeling (2022)
3. Ma, C., Zhang, X.: Gf-vae: A flow-based variational autoencoder for molecule generation. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp. 1181–1190 (2021)
4. Prykhodko, O., Johansson, S.V., Kotsias, P.-C., Arús-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)
5. De Cao, N., Kipf, T.: Molgan: An implicit generative model for small molecular graphs. arXiv preprint arXiv:1805.11973 (2018)
6. Gupta, A., Müller, A.T., Huisman, B.J., Fuchs, J.A., Schneider, P., Schneider, G.: Generative recurrent networks for de novo drug design. Molecular informatics **37**(1-2), 1700111 (2018)
7. Goel, M., Raghunathan, S., Laghuvarapu, S., Priyakumar, U.D.: Molegular: Molecule generation using reinforcement learning with alternating rewards. Journal of Chemical Information and Modeling **61**(12), 5815–5826 (2021)
8. Zang, C., Wang, F.: Moflow: an invertible flow model for generating molecular graphs. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 617–626 (2020)
9. Frey, N.C., Gadepally, V., Ramsundar, B.: Fastflows: Flow-based models for molecular graph generation. arXiv preprint arXiv:2201.12419 (2022)
10. Goldberg, Y.: A primer on neural network models for natural language processing. Journal of Artificial Intelligence Research **57**, 345–420 (2016)
11. Reverdy, P., Srivastava, V., Leonard, N.E.: Satisficing in multi-armed bandit problems. IEEE Transactions on Automatic Control **62**(8), 3788–3803 (2016)
12. Yasonik, J.: Multiobjective de novo drug design with recurrent neural networks and nondominated sorting. Journal of Cheminformatics **12**(1), 1–9 (2020)
13. Santana, M.V., Silva-Jr, F.P.: De novo design and bioactivity prediction of sars-cov-2 main protease inhibitors using recurrent neural network-based transfer learning. BMC chemistry **15**(1), 1–20 (2021)
14. Tong, X., Liu, X., Tan, X., Li, X., Jiang, J., Xiong, Z., Xu, T., Jiang, H., Qiao, N., Zheng, M.: Generative models for de novo drug design. Journal of Medicinal Chemistry **64**(19), 14011–14027 (2021)
15. Flam-Shepherd, D., Zhu, K., Aspuru-Guzik, A.: Language models can learn complex molecular distributions. Nature Communications **13**(1), 1–10 (2022)
16. Xiong, J., Xiong, Z., Chen, K., Jiang, H., Zheng, M.: Graph neural networks for automated de novo drug design. Drug Discovery Today **26**(6), 1382–1393 (2021)
17. Renz, P., Van Rompaey, D., Wegner, J.K., Hochreiter, S., Klambauer, G.: On failure modes in molecule generation and optimization. Drug Discovery Today: Technologies **32**, 55–63 (2019)
18. Chen, B., Wang, T., Li, C., Dai, H., Song, L.: Molecule optimization by explainable evolution. In: International Conference on Learning Representation (ICLR) (2021)
19. Sun, M., Xing, J., Meng, H., Wang, H., Chen, B., Zhou, J.: Molsearch: Search-based multi-objective molecular generation and property optimization. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 4724–4732 (2022)
20. 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
21. Mukaidaisi, M., Vu, A., Grantham, K., Tchagang, A., Li, Y.: Multi-objective drug design based on graph-fragment molecular representation and deep evolutionary learning. Frontiers in Pharmacology **13** (2022)
22. Chen, B., Wang, T., Li, C., Dai, H., Song, L.: Molecule optimization by explainable evolution. In: International Conference on Learning Representation (ICLR) (2021)
23. Krenn, M., Häse, F., Nigam, A., Friederich, P., Aspuru-Guzik, A.: Self-referencing embedded strings (selfies): A 100% robust molecular string representation. Machine Learning: Science and Technology **1**(4), 045024 (2020)
24. 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)
25. Rong, X.: word2vec parameter learning explained. arXiv preprint arXiv:1411.2738 (2014)
26. Nigam, A., Pollice, R., Krenn, M., dos Passos Gomes, G., Aspuru-Guzik, A.: Beyond generative models: superfast traversal, optimization, novelty, exploration and discovery (stoned) algorithm for molecules using selfies. Chemical science **12**(20), 7079–7090 (2021)
27. Mendez, D., Gaulton, A., Bento, A.P., Chambers, J., De Veij, M., Félix, E., Magariños, 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)
28. Landrum, G., et al.: RDKit: A software suite for cheminformatics, computational chemistry, and predictive modeling. Academic Press Cambridge (2013)
29. Vogt, M.: Using deep neural networks to explore chemical space. Expert Opinion on Drug Discovery **17**(3), 297–304 (2022)
30. O'Boyle, N., Dalke, A.: Deepsmiles: an adaptation of smiles for use in machine-learning of chemical structures (2018)

31. Popova, M., Isayev, O., Tropsha, A.: Deep reinforcement learning for de novo drug design. Science advances **4**(7), 7885 (2018)
32. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
33. Graves, A.: Long short-term memory. Supervised sequence labelling with recurrent neural networks, 37–45 (2012)
34. Abdi, H., Williams, L.J.: Principal component analysis. Wiley interdisciplinary reviews: computational statistics **2**(4), 433–459 (2010)
35. Bajusz, D., G Ferenczy, G., M Keseru, G.: Structure-based virtual screening approaches in kinase-directed drug discovery. Current topics in medicinal chemistry **17**(20), 2235–2259 (2017)
36. Yang, J.-M., Chen, C.-C.: Gemdock: a generic evolutionary method for molecular docking. Proteins: Structure, Function, and Bioinformatics **55**(2), 288–304 (2004)
37. Voršilák, M., Svozil, D.: Nonpher: computational method for design of hard-to-synthesize structures. Journal of cheminformatics **9**(1), 1–7 (2017)
38. Gao, W., Coley, C.W.: The synthesizability of molecules proposed by generative models. Journal of chemical information and modeling **60**(12), 5714–5723 (2020)
39. Olivecrona, M., Blaschke, T., Engkvist, O., Chen, H.: Molecular de-novo design through deep reinforcement learning. Journal of cheminformatics **9**(1), 1–14 (2017)
40. Thiede, L.A., Krenn, M., Nigam, A., Aspuru-Guzik, A.: Curiosity in exploring chemical spaces: intrinsic rewards for molecular reinforcement learning. Machine Learning: Science and Technology **3**(3), 035008 (2022)
41. Bradshaw, J., Paige, B., Kusner, M.J., Segler, M., Hernández-Lobato, J.M.: A model to search for synthesizable molecules. Advances in Neural Information Processing Systems **32** (2019)
42. 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: International Conference on Machine Learning, pp. 3668–3679 (2020). PMLR
43. Agyemang, B., Wu, W.-P., Addo, D., Kpiebaareh, M.Y., Nanor, E., Roland Haruna, C.: Deep inverse reinforcement learning for structural evolution of small molecules. Briefings in Bioinformatics **22**(4), 364 (2021)
44. Gao, W., Fu, T., Sun, J., Coley, C.W.: Sample efficiency matters: A benchmark for practical molecular optimization. arXiv preprint arXiv:2206.12411 (2022)