

# MARS: MARKOV MOLECULAR SAMPLING FOR MULTI-OBJECTIVE DRUG DISCOVERY

Yutong Xie<sup>†◊</sup>, Chence Shi<sup>†△</sup>, Hao Zhou<sup>†\*</sup>, Yuwei Yang<sup>†</sup>, Weinan Zhang<sup>‡</sup>, Yong Yu<sup>‡</sup>, Lei Li<sup>†\*</sup>

<sup>†</sup>ByteDance AI Lab, Shanghai, China

<sup>◊</sup>University of Michigan, Ann Arbor, MI, USA

<sup>△</sup>Montréal Institute of Learning Algorithms, Montreal, Canada

<sup>‡</sup>Department of Computer Science and Engineering, Shanghai Jiao Tong University, China

## ABSTRACT

Searching for novel molecules with desired chemical properties is crucial in drug discovery. Existing work focuses on developing neural models to generate either molecular sequences or chemical graphs. However, it remains a big challenge to find novel and diverse compounds satisfying several properties. In this paper, we propose MARS, a method for multi-objective drug molecule discovery. MARS is based on the idea of generating the chemical candidates by iteratively editing fragments of molecular graphs. To search for high-quality candidates, it employs Markov chain Monte Carlo sampling (MCMC) on molecules with an annealing scheme and an adaptive proposal. To further improve sample efficiency, MARS uses a graph neural network (GNN) to represent and select candidate edits, where the GNN is trained on-the-fly with samples from MCMC. Experiments show that MARS achieves state-of-the-art performance in various multi-objective settings where molecular bio-activity, drug-likeness, and synthesizability are considered. Remarkably, in the most challenging setting where all four objectives are simultaneously optimized, our approach outperforms previous methods significantly in comprehensive evaluations. The code is available at <https://github.com/yutxie/mars>.

## 1 INTRODUCTION

Drug discovery aims to find chemical compounds with desired target properties, such as high drug-likeness (Bickerton et al., 2012, QED). The problem is also referred to as molecular design, molecular generation, or molecular search. The space of drug-like chemicals is enormous, approximate  $10^{33}$  for realistic drugs that could ever be synthesized (Polishchuk et al., 2013). Therefore it is very challenging to search for high-quality molecules from such a vast space — enumeration would take almost forever. For a particular disease, finding the right candidates targeting specific proteins further complicates the problem.

Instead of enumerating or searching from the immense chemical space, recent work utilizes deep generative models to generate candidate molecules directly (Schwalbe-Koda & Gómez-Bombarelli, 2020). However, most prior work focuses on generating molecules concerning a single property such as drug-likeness (QED) or octanol-water partition coefficient (logP) (Jin et al., 2018; You et al., 2018; Popova et al., 2019; Shi et al., 2020; Zang & Wang, 2020). While in practical settings, typical drug discovery requires consideration of multiple properties jointly (Nicolaou et al., 2012). For example, to find drug-like molecules that are easy to synthesize and exhibit high biological activity against the target protein. Naturally, multi-objective molecule design is much more challenging than the single-objective scenario (Jin et al., 2020).

This paper studies the problem of multi-objective molecule design for drug discovery. An ideal solution should be efficient and meet the following criteria. *C1*: It should satisfy multiple properties with high scores; *C2*: It should produce novel and diverse molecules; *C3*: Its generation process does not

\*Work was done while Yutong Xie and Chence Shi were research interns at ByteDance. Corresponding to: zhouhao.nlp@bytedance.com and lileilab@bytedance.com.

rely on either expert annotated or wet experimental data collected from a biochemistry lab (since it requires tremendous effort and hard to obtain). Existing molecule generation approaches are mainly designed for the single objective setting, and they could not meet all criteria in the setting of multiple objectives. These methods belong to four categories: a) generating candidates from a learned continuous latent space (Gómez-Bombarelli et al., 2018; Jin et al., 2018), b) through reinforcement learning (You et al., 2018), c) using an encoder-decoder translation approach (Jin et al., 2019), or d) optimizing molecular properties through genetic algorithms (Nigam et al., 2020). Current state-of-the-art multi-objective molecular generation is a rationale-based method (Jin et al., 2020). In this approach, the authors propose to build molecules by composing multiple extracted rationales, and the model can generate compounds that are simultaneously active to multiple biological targets. However, such an approach will result in quite complex molecules when we have many objectives. This is because different objectives correspond to different rationales, and including all these rationales could lead to large molecules, which may be less drug-like and hard to be synthesized practically.

In this paper, we propose MARKov molecular SAMpling (MARS), a simple yet flexible method for drug discovery. The basic idea is to start from a seed molecule and keep generating candidate molecules by modifying fragments of molecular graphs from previous steps. It meets all the criteria C1-3. In MARS, the molecular design is formulated as an iterative editing procedure with its total objective consisting of multiple property scores (C1). MARS employs the annealed Markov chain Monte Carlo sampling method to search for optimal chemical compounds, which allows for the exploration of chemicals with novel and different fragments (C2). The proposal to modify molecular fragments is represented using graph neural networks (GNNs), whose parameters are adaptively learned. We used message passing neural networks (MPNNs) in practice (Gilmer et al., 2017), but other GNNs can fit the framework as well. Furthermore, MARS utilizes the sample paths generated on-the-fly to train the proposal network adaptively. Therefore, it does not rely on external annotated data (C3). With such an adaptive learnable proposal, it keeps improving the generation quality throughout the process.

We evaluate MARS and four other baselines, one latest method for each of the four method categories. The benchmark includes a variety of multi-objective generation settings. Experiments show that our proposed MARS achieves state-of-the-art performance on five out of six tasks in terms of a comprehensive evaluation consisting of the success rate, novelty, and diversity of the generated molecules. Notably, in the most challenging setting where four objectives – bio-activities to two different targets, drug-likeness, and synthesizability – are simultaneously considered, our method achieves the state-of-the-art result and outperforms existing methods by 77% in the comprehensive evaluation.

Our contributions are as follows:

- We present MARS, a generic formulation of molecular design using Markov sampling, which can easily accommodate multiple objectives.
- We develop an adaptive fragment-editing proposal based on GNN that is learnable on the fly with only samples self-generated and efficient in exploring the chemical space.
- Experiments verifies our proposed MARS framework can find novel and diverse bioactive molecules that are both drug-like and highly synthesizable.

## 2 RELATED WORK

Recent years have witnessed the success of applying deep generative models and molecular graph representation learning in drug discovery (Schwalbe-Koda & Gómez-Bombarelli, 2020; Guo & Zhao, 2020). Existing approaches for molecular property optimization can be grouped into four categories, including generation with a) Bayesian inference, b) reinforcement learning, c) encoder-decoder translation models, and d) evolutionary and genetic algorithms. The first category is learning continuous latent spaces for molecular sequences or graphs and generating from such spaces using Bayesian optimization (BO) (Gómez-Bombarelli et al., 2018; Jin et al., 2018; Winter et al., 2019). These methods rely heavily on the quality of latent representations, which imposes huge challenges to the encoders when there are multiple properties to consider.

Unlike the first class, other work uses reinforcement learning (RL) to optimize desired objectives directly in the explicit chemical space (De Cao & Kipf, 2018; Popova et al., 2018; You et al., 2018;

Popova et al., 2019; Shi et al., 2020). However, the models are usually hard to train due to the high variance of RL.

The third category directly trains a translation model that maps from an input molecule to a high-quality output molecule (Jin et al., 2019; 2020). Although simple, such methods require many high-quality labeled data, making them impractical in scenarios where the data is limited.

The last category of methods are evolutionary algorithms (EAs) and genetic algorithms (GAs) to explore large chemical space with certain property (Nicolaou et al., 2012; Devi et al., 2015; Jensen, 2019; Ahn et al., 2020). In Nigam et al. (2020), the authors propose to augment GA by adding an adversarial loss into the fitness evaluation to increase the diversity, and the augmented GA outperforms all other generative models in optimizing logP. Though flexible and straightforward, to make the search process efficient enough, most GA and EA methods require domain experts to design molecular mutation and crossover rules, which could be non-trivial to obtain.

Besides single property optimization, there is recent work to address the multi-objective molecule generation problem. For example, Li et al. (2018) proposes to use a conditional generative model to incorporate several objectives flexibly, while Lim et al. (2020) leverages molecular scaffolds to control the properties of generated molecules better. Among them, the current state-of-the-art approach is a rationale-based method proposed by Jin et al. (2020). In this method, the authors propose to build molecules by assembling extracted rationales. Despite its great success in generating compounds simultaneously active to multiple biological targets, the combination of rationales might hinder the synthesizability and drug-likeness of produced molecules, as they tend to be large as the number of objectives grows. In contrast, our MARS framework turns the generation problem into a sampling procedure, which serves as an alternative way compared with deep generative models, and can efficiently discover bio-active molecules that are both drug-like and highly synthesizable.

Remotely related is recent work to generate molecules through sampling. Seff et al. (2019) defines a Gibbs sampling procedure, in which the Markov chain alternates between randomly corrupting the molecules and recovering the corrupted ones with a learned reconstruction model. However, this method mainly focuses on generating molecules that follow the observed data distribution and cannot be directly tailored for property optimization. Different from this work, MARS is built upon the general MCMC sampling framework, which allows further enhancement with adaptive proposal learning to edit molecular graphs efficiently. Actually, generating instances from a discrete space with MCMC sampling methods is previously employed in various other applications, e.g., generating natural language sentences under various constraints (Miao et al., 2019; Zhang et al., 2019; Liu et al., 2020; Zhang et al., 2020).

### 3 PROPOSED MARS APPROACH

In this section, we present the MArkov molecular Sampling method (MARS) for multi-objective molecular design. We define a Markov chain over the explicit molecular graph space and design a kernel to navigate high probable candidates with acceptance and rejection.

#### 3.1 SAMPLING FROM THE MOLECULAR SPACE

Our proposed MARS framework aims at sampling molecules with desired properties from the chemical space. Specifically, given  $K$  properties of interest, the desired molecular distribution can be formulated as a combination of all objectives:

$$\pi(x) = \underbrace{s_1(x) \circ s_2(x) \circ s_3(x) \circ \dots \circ s_K(x)}_{\text{desired properties}} \quad (1)$$

where  $x$  is a molecule in the molecular space  $\mathcal{X}$ .  $\pi(x)$  is an unnormalized distribution over molecules integrating the desired properties.  $s_k(x)$  is a scoring function for the  $k$ -th property and the “ $\circ$ ” operator stands for a combination of scores (e.g., summation or multiplication). In practical drug discovery, these terms could be related to the biological activity, drug-likeness, and synthesizability of molecules (Nicolaou et al., 2012). This framework allows flexible configuration according to various concrete applications. However, as the number of objectives grows, the joint distribution  $\pi(x)$  will become more complex and intractable, making the sampling non-trivial.

In MARS, we propose to sample molecules from the desired distribution Eq. 1 using Markov chain Monte Carlo (MCMC) methods (Andrieu et al., 2003). Given a desired molecular distribution  $\pi(x)$  as the unnormalized target distribution, we define a Markov chain on the explicit chemical space  $\mathcal{X}$  (i.e., each state of the Markov chain is a particular molecule) and introduce a proposal distribution  $q(x' | x)$  to perform state transitions.

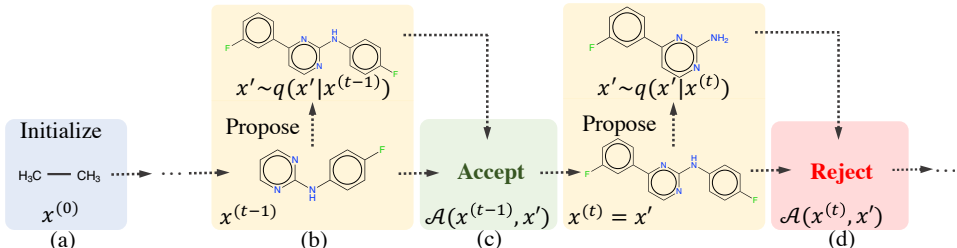


Figure 1: The framework of MARS. During the sampling process: (a) starting from an arbitrary initial molecule  $x^{(0)}$  in the molecular space  $\mathcal{X}$ , (b) sampling a candidate molecule  $x' \in \mathcal{X}$  from the proposal distribution  $q(x' | x^{(t-1)})$  at each step, and (c/d) the candidate  $x'$  is either accepted or rejected according to the acceptance rate  $\mathcal{A}(x^{(t-1)}, x') \in [0, 1]$ . By repeating this process, we can generate a sequence of molecules  $\{x^{(t)}\}_{t=0}^{\infty}$ .

Specifically, as shown in Figure 1, the sampling procedure of MARS starts from an initial molecule  $x^{(0)} \in \mathcal{X}$ . At each time step  $t$ , a molecule candidate  $x' \in \mathcal{X}$  will be sampled from the proposal distribution  $q(x' | x^{(t-1)})$ , where  $x^{(t-1)}$  denotes the molecule at time step  $t - 1$ . Then the proposed candidate  $x'$  could be either accepted  $x^{(t)} = x'$  or rejected  $x^{(t)} = x^{(t-1)}$  according to the acceptance rate  $\mathcal{A}(x^{(t-1)}, x') \in [0, 1]$  controlled by the target distribution  $\pi(x)$ . By repeating this process, a sequence of molecules  $\{x^{(t)}\}_{t=0}^{\infty}$  can be generated. Such sequence of molecules will converge to the target distribution  $\pi(x)$  if the proposal distribution and the acceptance mechanism are configured properly.

The acceptance rate is calculated as follow:

$$\mathcal{A}(x, x') = \min \left\{ 1, \frac{\pi^\alpha(x')q(x|x')}{\pi^\alpha(x)q(x'|x)} \right\} \quad (2)$$

where  $\alpha$  is a coefficient that varies in different instantiations of MCMC algorithms. Here to find molecules that globally maximize the target distribution, we employ an annealing scheme (Laarhoven & Aarts, 1987) where  $\alpha = 1/T$  and  $T$  is a temperature controlled by a cooling schedule. In addition to this, other instantiations such as Metropolis-Hastings (MH) algorithm (Metropolis et al., 1953) where  $\alpha = 1$  are also feasible under our general framework.

As for the proposal distribution  $q(x' | x)$ , it largely affects the sampling performance and should be designed elaborately. In general, it is crucial that the proposal distribution  $q(x' | x)$  and the target distribution  $\pi(x')$  are as close as possible to ensure high sampling efficiency. So we propose using a proposal distribution  $q_\theta(x' | x)$  with learnable parameters to capture the desired molecular properties and develop a strategy to train the proposal throughout the sampling process adaptively. The details will be described in the next section.

### 3.2 ADAPTIVE MOLECULAR GRAPH EDITING PROPOSAL

In this section we will examine in detail our proposed adaptive proposal distribution  $q_\theta(x' | x)$ . A molecule is represented as a graph whose nodes are heavy atoms and edges are chemical bonds. The proposal distribution is defined over molecular graph editing actions. We use the message passing neural network (MPNN) to represent the proposal. Alternative parameterization schemes such as other graph neural networks are also possible. To sample molecules with desired properties effectively and efficiently, we also design a self-training strategy to learn the proposal MPNN during sampling in an adaptive manner.

**Molecular graph editing actions.** To transform a molecule  $x$  into another molecule  $x'$ , we consider two sets of graph editing actions, i.e., fragment *adding* and *deleting*. These actions are inspired by fragment-based drug design (FBDD) methodology, whose success in drug discovery has been proved in past decades (Kumar et al., 2012). In MARS, we define *fragments* as connected components in

molecules separated by *single* bonds. To reduce the complexity of editing actions, we only consider fragments with a single attachment position. Moreover, we also define a *fragment vocabulary* that contains finitely many fragments, and only fragments in the vocabulary are allowed to be added onto a molecule. Examples for fragment adding and deleting actions are shown in Figure 2.

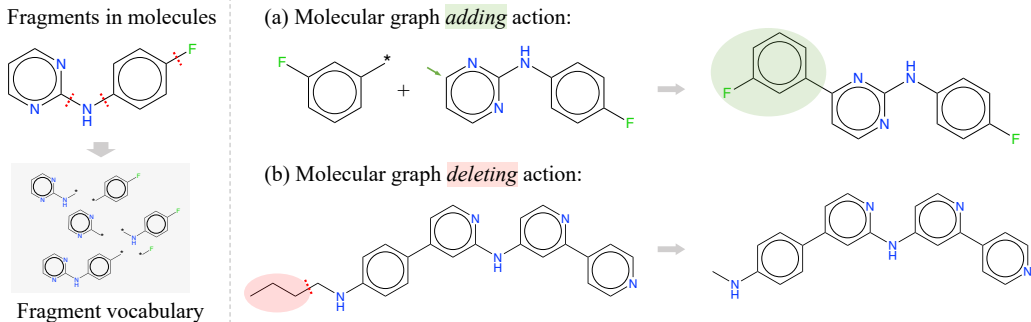


Figure 2: **Left:** Examples of molecular fragments and a fragment vocabulary. Red dashed lines represents cuttable bonds to extract fragments. **Right:** Examples of molecular graph editing actions.

Specifically, given a molecule  $x$  with  $n$  atoms and  $m$  bonds, we choose to add or delete a fragment onto or from this molecule randomly with probability  $\frac{1}{2}$  for each set of actions. For the *adding* action, suppose we have a probability distribution over atoms  $p_{\text{add}}(x, u)$  and a probability distribution over fragments in the vocabulary  $p_{\text{frag}}(x, u, k)$ . Here  $u \in [n]$  is an indicator of the atom in  $x$  to which the fragment is adding to and  $k \in [V]$  is an indicator of fragments in the vocabulary of size  $V$ . We can compute the proposal distribution as follows:

$$q(x'|x) = \frac{1}{2} \cdot p_{\text{add}}(x, u) \cdot p_{\text{frag}}(x, u, k) \quad (3)$$

where  $x'$  is the molecule obtained by adding the  $k$ -th fragment onto the atom  $u$  in molecule  $x$ .

As for the *deleting* action, suppose we have a probability distribution over bonds<sup>1</sup>  $p_{\text{del}}(x, b)$  where  $b \in [2m]$  is an indicator of bonds in  $x$ . We can compute the proposal distribution as follow:

$$q(x'|x) = \frac{1}{2} \cdot p_{\text{del}}(x, b) \quad (4)$$

where  $x'$  is the molecule obtained by removing bond  $b$  and the attached fragment from molecule  $x$ .

**Parameterizing with MPNNs.** To better model the molecular graph editing actions, we propose to use MPNNs to suggest the probability distributions  $(p_{\text{add}}, p_{\text{frag}}, p_{\text{del}}) = \mathcal{M}_{\theta}(x)$  where  $\mathcal{M}_{\theta}$  is a MPNN model specified by parameters  $\theta$ , which has been proven powerful to predict chemical properties with molecular graphs (Gilmer et al., 2017). Given a molecule  $x$ , we compute the probability distributions as follow:

$$\mathbf{h}_u^{\text{node}} = \text{MPNN}(x)_u \in \mathbb{R}^d \quad (5)$$

$$\mathbf{h}_b^{\text{edge}} = \text{Concat}(\mathbf{h}_v^{\text{node}}, \mathbf{h}_w^{\text{node}}) \in \mathbb{R}^{2d} \quad (6)$$

$$p_{\text{add}}(x) = \text{Softmax}(\{\text{MLP}_{\text{node}}(\mathbf{h}_u^{\text{node}})\}_{u=1}^n) \in [0, 1]^n \quad (7)$$

$$p_{\text{frag}}(x, u) = \text{Softmax}(\text{MLP}'_{\text{node}}(\mathbf{h}_u^{\text{node}})) \in [0, 1]^{|V|} \quad (8)$$

$$p_{\text{del}}(x) = \text{Softmax}(\{\text{MLP}_{\text{edge}}(\mathbf{h}_b^{\text{edge}})\}_{b=1}^{2m}) \in [0, 1]^{2m} \quad (9)$$

where  $u$  is an atom indicators,  $\{\mathbf{h}_u^{\text{node}}\}_{u=1}^n$  are node hidden representations,  $v, w$  are atoms connected with bond  $b$ ,  $\{\mathbf{h}_b^{\text{edge}}\}_{b=1}^{2m}$  are edge hidden representations, and  $\text{MLP}_{\text{node}}, \text{MLP}'_{\text{node}}, \text{MLP}_{\text{edge}}$  are multi-layer perceptrons (MLPs), similar to Hu et al. (2020).

**Adaptive self-training.** To capture the desired properties and improve the sampling effectiveness, we can train the editing model to increase the probability of suggesting high-quality candidate

<sup>1</sup>Molecular bonds are treated as directional to specify the fragments to drop from molecules.

**Algorithm 1:** MARS

---

```

1 Set  $N$  initial molecules  $\{x_i^{(0)}\}_{i=1}^N$  and initialize the molecular graph editing model  $\mathcal{M}_\theta$ 
2 Create an empty editing model training dataset  $\mathcal{D} = \{\}$ 
3 for  $t = 1, 2, \dots$  do
4   for  $i = 1, 2, \dots, N$  do
5     Compute probability distributions  $(p_{\text{add}}, p_{\text{frag}}, p_{\text{del}}) = \mathcal{M}_\theta(x_i^{(t-1)})$  as Equations 7-9
6     Sample a candidate molecule  $x'$  from the proposal distribution  $q(x' | x_i^{(t-1)})$  defined with
       probability distributions  $p_{\text{add}}, p_{\text{frag}}, p_{\text{del}}$  as Equations 3-4
7     if  $u < \mathcal{A}(x_i^{(t-1)}, x')$  where  $u \sim \mathcal{U}_{[0,1]}$  then
8       | Accept the candidate molecule  $x_i^{(t)} = x'$ 
9     else
10      | Refuse the candidate molecule  $x_i^{(t)} = x_i^{(t-1)}$ 
11     if The candidate improves the objectives, i.e.  $\pi(x') > \pi(x_i^{(t-1)})$  then
12      | Adding the editing record  $(x_i^{(t-1)}, x')$  into the dataset  $\mathcal{D}$ 
13    $\theta^{\text{new}} \leftarrow \arg \max \log M_\theta(\mathcal{D})$ 

```

---

molecules. Here we propose to train the model on-the-fly during the sampling process in an adaptive manner where the training data is collected from the sampling paths. By doing so, we can bypass the difficulty of lacking training instances that satisfy all property constraints. Mainly, we collect molecule candidates that improve our desired objectives and train the model  $\mathcal{M}_\theta$  in a maximum likelihood estimation (MLE) manner (i.e., to maximize the probability of producing the collected candidates). The overall MARS is described in Algorithm 1.

**Discussion on convergence.** Compared with standard MCMC algorithms, MARS still falls in the Metropolis-Hastings algorithm but with an annealing scheme and an adaptive proposal, which results in inhomogeneous transition kernels. The convergence of adaptive MCMC is discussed in Rosenthal (2011). According to the diminishing adaptation condition, we can ensure convergence by making the difference of proposals in consecutive iterations diminish to zero. MARS can satisfy this condition by using an optimizer whose learning rate will shrink to zero eventually (e.g., Adam). Annealed MCMC is to find samples maximizing the target probability. The convergence of annealed MCMC is discussed in Andrieu et al. (2003).

## 4 EXPERIMENTS

### 4.1 EXPERIMENT SETUP

**Biological objectives.** Following Jin et al. (2020), we consider the following inhibition scores against two Alzheimer-related target proteins as the biological activity objectives. The score is given by a random forest model<sup>2</sup> that predicts based on Morgan fingerprint features of a molecule (Rogers & Hahn, 2010).

- GSK3 $\beta$ : Inhibition against glycogen synthase kinase-3 $\beta$ .
- JNK3: Inhibition against c-Jun N-terminal kinase-3.

**Non-biological objectives.** Following Jin et al. (2020), we adopt QED (Bickerton et al., 2012) and synthetic accessibility (SA) (Ertl & Schuffenhauer, 2009) to quantify the drug-likeness and synthesizability. We rescale the SA score (initially between 10 and 1) into  $[0, 1]$  such that molecules with higher scores are more synthesizable.

**Multi-objective generation setting.** To evaluate the effectiveness of the proposed method for multi-objective drug design, we also consider the following more challenging objective combinations:

<sup>2</sup><https://github.com/wengong-jin/multiobj-rationale>

- GSK3 $\beta$ +JNK3: Jointly inhibiting GSK3 $\beta$  and JNK3. The combination may provide potential benefits for the treatment of Alzheimer’s disease reported by Hu et al. (2009); Martin et al. (2013).
- GSK3 $\beta$ /JNK3+QED+SA: Inhibiting GSK3 $\beta$  or JNK3 while being drug-like and synthetically accessible, which are quantified by QED and SA, respectively.
- GSK3 $\beta$ +JNK3+QED+SA: Jointly inhibiting GSK3 $\beta$  and JNK3 while being drug-like and synthetically accessible, which are quantified by QED and SA, respectively.

**Baselines.** We compare MARS with the following methods – the latest ones from four categories mentioned in the related work (Sec. 2). **GCPN** (You et al., 2018) leverages RL to generate molecules atom by atom, and the adversarial loss is incorporated in the objective to generate more realistic molecules. **JT-VAE** (Jin et al., 2018) is a VAE-based approach that firstly generates junction trees and then assembles them into molecules. It performs Bayesian optimization (BO) to guide molecules towards desired properties. **RationaleRL** (Jin et al., 2020) is a state-of-the-art approach for multi-property optimization, which generates molecules from combined rationales. **GA+D** (Nigam et al., 2020) is a heuristic search method that applies the genetic algorithm (GA) to find molecules with high property scores. An adversarial loss is incorporated in the fitness evaluation to increase the diversity of generated molecules.

**Evaluation metrics.** Following Jin et al. (2020), we generate  $N = 5000$  molecules for each approach and compare the proposed method with the baselines on the following evaluation metrics: **Success rate (SR)** is the percentage of generated molecules that are evaluated as positive on all given objectives (QED  $\geq 0.6$ , SA  $\geq 0.67$ , the inhibition scores of GSK3 $\beta$  and JNK3  $\geq 0.5$ ); **Novelty (Nov)** is the percentage of generated molecules with similarity less than 0.4 compared to the nearest neighbor  $x_{\text{SNN}}$  in the training set (Olivecrona et al., 2017):  $\text{Nov} = \frac{1}{n} \sum_{x \in \mathcal{G}} \mathbf{1}[\text{sim}(x, x_{\text{SNN}}) < 0.4]$ ; **Diversity (Div)** measures the diversity of generated molecules, which can be calculated based on pairwise Tanimoto similarity over Morgan fingerprints  $\text{sim}(x, x')$  as  $\text{Div} = \frac{2}{n(n-1)} \sum_{x \neq x' \in \mathcal{G}} 1 - \text{sim}(x, x')$ ; **PM** is the product of the above three metrics, which is a more comprehensive evaluation of the proposed method. Intuitively, PM presents the percentage of generated molecules that are simultaneously bio-active, novel and diverse, which are essential criteria for molecules to be considered in building a suitable drug candidate library in early-stage drug discovery (Huggins et al., 2011).

**Implementation details.** For the fragment vocabulary, we extract the top 1000 frequently appearing fragments that contain no more than 10 heavy atoms from the ChEMBL database (Gaulton et al., 2017) by enumerating single bonds to break. As for the sampling process, the unnormalized target distribution is set as  $\pi(x) = \sum_k s_k(x)$  where  $s_k(x)$  is a scoring function for the above-mentioned properties of interests, the temperature is set as  $T = 0.95^{\lfloor t/5 \rfloor}$  and we sample  $N = 5000$  molecules at one time. During sampling, the computation of  $q(x | x')$  is ignored and we approximate  $\mathcal{A}(x, x')$  with  $\min\{1, \pi^\alpha(x')/\pi^\alpha(x)\}$  to increase the computation efficiency. This is acceptable because in practice  $q(x | x')$  and  $q(x' | x)$  is of order  $O(1)$  and  $\mathcal{A}(x, x')$  will be gradually bounded by  $\pi^\alpha(x')/\pi^\alpha(x)$  as the temperature  $T$  decrease to zero. The sampling paths are all starting with an identical molecule “C-C”, which is also adopted by previous graph generation methods for organic molecules (You et al., 2018). The MPNN model has six layers, and the node embedding size is  $d = 64$ . Moreover, for the model training, we use an Adam optimizer (Kingma & Ba, 2015) to update the model parameters with an initial learning rate set as  $3 \times 10^{-4}$ , the maximum dataset size is limited as  $|\mathcal{D}| \leq 75,000$ , and at each step, we update the model for no more than 25 times.

## 4.2 MAIN RESULTS AND ANALYSIS

We perform ten independent runs for MARS. The quantitative results are summarized in Table 1 and Table 2. From these tables, we observe that MARS outperforms all the baselines on five out of six tasks in terms of PM. Furthermore, on the most challenging multi-objective optimization task, i.e., GSK3 $\beta$ +JNK3+QED+SA, it significantly surpasses the best baseline with a 77% improvement for the product of metrics PM. Additional results are shown in Appendix A.

In comparing all these methods, the GA+D baseline is most similar to our MARS in terms of the high novelty and PM score, as both methods focus on molecular space exploration. However, the diversity score of GA+D drops a lot when optimizing multiple properties simultaneously, as GAs are likely to get trapped in regions of local optima (Paszkowicz, 2009). RationaleRL is a very strong

Table 1: Comparison of different methods on molecular generation with only bio-activity objectives. Results of GA+D are obtained by running its open-source code. Results of other baselines are taken from Jin et al. (2020). For MARS, we report the mean and standard deviation of 10 independent experiments.

Method	GSK3 $\beta$				JNK3				GSK3 $\beta$ + JNK3			
	SR	Nov	Div	PM	SR	Nov	Div	PM	SR	Nov	Div	PM
GCPN	42.4%	11.6%	0.904	0.04	32.3%	4.4%	0.884	0.01	3.5%	8.0%	0.874	0.00
JT-VAE	32.2%	11.8%	0.901	0.03	23.5%	2.9%	0.882	0.01	3.3%	7.9%	0.883	0.00
RationaleRL	100.0%	53.4%	0.888	0.47	100.0%	46.2%	0.862	0.40	100.0%	97.3%	0.824	<b>0.80</b>
GA+D	84.6%	100.0%	0.714	<b>0.60</b>	52.8%	98.3%	0.726	0.38	84.7%	100.0%	0.424	0.36
MARS	100.0%	84.0%	0.718	<b>0.60</b> $\pm 0.04$	98.8%	88.9%	0.748	<b>0.66</b> $\pm 0.04$	99.5%	75.3%	0.691	0.52 $\pm 0.08$

Table 2: Comparison of different methods on molecular generation with bio-activity, QED, and SA objectives. Results of all baselines are obtained by running their open-source codes. For the results of MARS, we report the mean and standard deviation of 10 independent experiments.

Method	GSK3 $\beta$ + QED + SA				JNK3 + QED + SA				GSK3 $\beta$ + JNK3 + QED + SA			
	SR	Nov	Div	PM	SR	Nov	Div	PM	SR	Nov	Div	PM
GCPN	0.0%	0.0%	0.000	0.00	0.0%	0.0%	0.000	0.00	0.0%	0.0%	0.000	0.00
JT-VAE	9.6%	95.8%	0.680	0.06	21.8%	<b>100.0%</b>	0.600	0.13	5.4%	<b>100.0%</b>	0.277	0.02
RationaleRL	69.9%	40.2%	<b>0.893</b>	0.25	62.3%	37.6%	<b>0.865</b>	0.20	75.0%	55.5%	0.706	0.29
GA+D	89.1%	<b>100.0%</b>	0.682	0.61	85.7%	99.8%	0.504	0.43	85.7%	<b>100.0%</b>	0.363	0.31
MARS	<b>99.5%</b>	95.0%	0.719	<b>0.68</b> $\pm 0.03$	<b>91.3%</b>	94.8%	0.779	<b>0.67</b> $\pm 0.02$	<b>92.3%</b>	82.4%	<b>0.719</b>	<b>0.55</b> $\pm 0.05$

baseline that performs better than MARS in the GSK3 $\beta$ +JNK3 setting. Nevertheless, when taking the drug-likeness and synthetic accessibility into consideration, their performance falls short of ours and fails to generate novel molecules. The performance of GCPN and JT-VAE remains relatively low in most settings, as they are not tailored for multi-objective property optimization.

**Visualization.** We use t-SNE (van der Maaten & Hinton, 2008) to visualize the distribution of generated positive molecules with the positive ones in the training set under the GSK3 $\beta$ +JNK3+QED+SA setting. In the visualization, we use the ECFP6 fingerprints as suggested in Li et al. (2018). As shown by Figure 3, most molecules generated by GA+D fall into two massive clusters, which aligns their low diversity. Molecules generated by RationaleRL also tend to be clustered, with each cluster standing for a specific combination of rationales. By contrast, the molecules generated by MARS are evenly distributed in the space with a range of novel regions covered, which justifies our high novelty and diversity scores. We further visualize some molecules generated by MARS with high property scores in Figure 4, indicating its ability to generate highly synthesizable drug-like molecules that jointly inhibit GSK3 $\beta$  and JNK3. Additional examples of sampled molecules are shown in Appendix C.

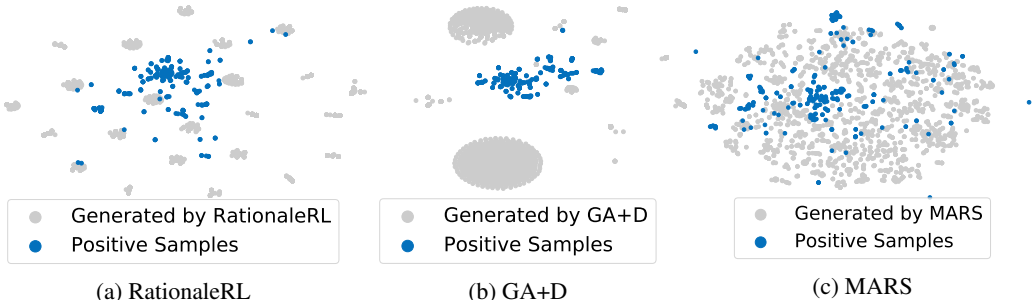


Figure 3: t-SNE visualization of generated molecules (gray) and positive molecules in the training set (blue).

**Running time.** The computing server has two CPUs with 64 virtual cores (2.10GHz), 231G memory (about 50G used), and one Tesla V100 GPU with 32G memory. In the GSK3 $\beta$ +JNK3+QED+SA setting, MARS takes roughly  $T = 550$  sampling steps and 12 hours in total to converge (including the time used in proposing and evaluating molecules as well as MPNN model training). For other baselines, RationaleRL takes 5.7 hours to fine-tune the model, and GA+D takes 278 steps and 2.2h to



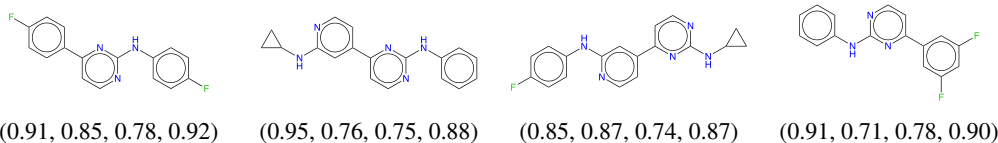


Figure 4: Sample molecules generated by MARS in the GSK3 $\beta$ +JNK3+QED+SA setting. The numbers in brackets are GSK3 $\beta$ , JNK3, QED, and SA scores of each molecule respectively.

achieve its best performance. Compared to the conventional drug discovery process, which usually takes months to years, the time we spent on molecular generation models is almost ignorable.

### 4.3 EFFECTS OF PROPOSAL AND ACCEPTANCE STRATEGY

To justify the contributions of the designed proposal and acceptance strategy, we compare them with some naive ones and summarize the results of different combinations in Table 3. For acceptance strategies, **Annealed** stands for annealed MCMC where the acceptance rate is computed as Equation 2 given  $\alpha = 1/T$ , **AlwaysAC** stands for always accepting the candidate, i.e.,  $\mathcal{A}(x, x') \equiv 1$ , and **HillClimb** stands for accepting the candidate only when the overall score is improved, i.e.,  $\mathcal{A}(x, x') = \text{sign}[s(x') > s(x)]$ . For proposal strategies, **Random** stands for random proposal where we randomly select atoms, bonds, and fragments to edit, and **Adaptive** stands for the adaptive fragment-based graph editing model trained during the sampling process as described in Section 3.2.

Table 3: Results of different acceptance strategies and proposal strategies for molecular sampling.

AC Strategy	Proposal	GSK3 $\beta$ + JNK3				GSK3 $\beta$ + JNK3 + QED + SA			
		SR	Nov	Div	PM	SR	Nov	Div	PM
Annealed	Random	40.9%	94.9%	0.828	0.32	25.5%	80.4%	0.793	0.16
AlwaysAC	Adaptive	49.1%	88.4%	0.742	0.32	10.1%	94.6%	0.716	0.07
HillClimb	Adaptive	53.7%	96.1%	0.814	0.42	51.4%	86.6%	0.777	0.35
Annealed	Adaptive	99.5%	75.2%	0.688	0.52	92.3%	82.4%	0.719	0.55

The results in Table 3 indicate that proposals will influence the performance of MARS dramatically (the first and the last row), especially when the number of objectives increases. The proposed adaptive proposal outperforms the random proposal and converges 4.6x faster in practice. By comparing the last three rows, we find the Annealed strategy outperforms the other two strategies by a large margin on both settings, as samples from such strategy are more likely to jump out of local optima. Another interesting observation is that even with the naive AlwaysAC or heuristic HillClimb strategy, the MARS achieves comparable or even better performance than GA+D and RationaleRL in some settings, e.g., HillClimb on GSK3 $\beta$ +JNK3+QED+SA optimization, which again proves the effectiveness of the proposed proposal.

## 5 CONCLUSION AND FUTURE WORK

This paper proposes a simple yet flexible MArkov moleculaR Sampling framework (MARS) for multi-objective drug discovery. MARS includes a trainable proposal to modify chemical graph fragments, which is parameterized by an MPNN. Our experiments verify that MARS outperforms prior approaches on five out of six molecule generation tasks, and it is capable of finding novel and diverse bioactive molecules that are both drug-like and highly synthesizable. Future work can include further study of parameterization and training strategy of the molecular-editing proposal.

## 6 ACKNOWLEDGEMENT

We would like to thank Meihua Dang for refactoring much of the MARS code. Meihua also performed multiple experiments, which generates the results for the tables. We also thank Jiangjie Chen, Yuxuan Song, Jingjing Xu, Weiying Ma, Hang Li, and anonymous reviewers for their constructive comments and suggestions.

## REFERENCES

- Sungsoo Ahn, Junsu Kim, Hankook Lee, and Jinwoo Shin. Guiding deep molecular optimization with genetic exploration. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems*, 2020.
- Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I. Jordan. An introduction to mcmc for machine learning. *Machine Learning*, 50(1):5–43, 2003.
- G. Richard Bickerton, Gaia V. Paolini, Jérémy Besnard, Sorel Muresan, and Andrew L. Hopkins. Quantifying the chemical beauty of drugs. *Nature Chemistry*, 4(2):90–98, 2012.
- Nicola De Cao and Thomas Kipf. MolGAN: An implicit generative model for small molecular graphs. *ICML 2018 workshop on Theoretical Foundations and Applications of Deep Generative Models*, 2018.
- R. Vasundhara Devi, S. Siva Sathya, and Mohane Selvaraj Coumar. Evolutionary algorithms for de novo drug design - a survey. *Applied Soft Computing*, 27:543–552, 2015.
- Peter Ertl and Ansgar Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of Cheminformatics*, 1(1):8–8, 2009.
- Anna Gaulton, Anne Hersey, Michał Nowotka, A. Patrícia Bento, Jon Chambers, David Mendez, Prudence Mutowo, Francis Atkinson, Louisa J. Bellis, Elena Cibrián-Uhalte, Mark Davies, Nathan Dedman, Anneli Karlsson, María Paula Magariños, John P. Overington, George Papadatos, Ines Smit, and Andrew R. Leach. The chEMBL database in 2017. *Nucleic Acids Research*, 45, 2017.
- Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.
- Xiaojie Guo and Liang Zhao. A systematic survey on deep generative models for graph generation. *arXiv: Learning*, 2020.
- Rafael Gómez-Bombarelli, Jennifer Nanshan Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan Prescott Adams, and Alan Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.
- Shuxin Hu, Aynun N. Begum, Mychica R. Jones, Mike S. Oh, Walter K. Beech, Beverly Hudspeth Beech, Fusheng Yang, Pingping Chen, Oliver J. Ubeda, Peter C. Kim, Peter Davies, Qiulan Ma, Greg M. Cole, and Sally A. Frautschy. Gsk3 inhibitors show benefits in an alzheimer’s disease (ad) model of neurodegeneration but adverse effects in control animals. *Neurobiology of Disease*, 33(2):193–206, 2009.
- Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay S. Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *Proceedings of the 8th International Conference on Learning Representations*, 2020.
- David John Huggins, Ashok Ramakrishnan Venkitaraman, and David Robert Spring. Rational methods for the selection of diverse screening compounds. *ACS Chemical Biology*, 6(3):208–217, 2011.
- Jan H. Jensen. A graph-based genetic algorithm and generative model/monte carlo tree search for the exploration of chemical space. *Chemical Science*, 10(12):3567–3572, 2019.
- Wengong Jin, Regina Barzilay, and Tommi S. Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.
- Wengong Jin, Kevin Yang, Regina Barzilay, and Tommi S. Jaakkola. Learning multimodal graph-to-graph translation for molecule optimization. In *Proceedings of the 7th International Conference on Learning Representations*, 2019.

- Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Composing molecules with multiple property constraints. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.
- Wengong Jin, Regina Barzilay, and Tommi S. Jaakkola. Hierarchical generation of molecular graphs using structural motifs. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.
- Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, 2015.
- Ashutosh Kumar, A Voet, and K. Y. J Zhang. Fragment based drug design: from experimental to computational approaches. *Current Medicinal Chemistry*, 19(30):5128–5147, 2012.
- P. J. M. Laarhoven and E. H. L. Aarts. *Simulated Annealing: Theory and Applications*. 1987.
- Yibo Li, Liang Ren Zhang, and Zhenming Liu. Multi-objective de novo drug design with conditional graph generative model. *Journal of Cheminformatics*, 10(1):33, 2018.
- Jaechang Lim, Sang-Yeon Hwang, Seungsu Kim, Seokhyun Moon, and Woo Youn Kim. Scaffold-based molecular design using graph generative model. *Chemical Science*, 11(4):1153–1164, 2020.
- Xianggen Liu, Lili Mou, Fandong Meng, Hao Zhou, Jie Zhou, and Sen Song. Unsupervised paraphrasing by simulated annealing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.
- Ludovic Martin, Xenia Latypova, Cornelia M. Wilson, Amandine Magnaudeix, Marie-Laure Perrin, Catherine Yardin, and Faraj Terro. Tau protein kinases: Involvement in alzheimer’s disease. *Ageing Research Reviews*, 12(1):289–309, 2013.
- Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.
- Ning Miao, Hao Zhou, Lili Mou, Rui Yan, and Lei Li. CGMH: constrained sentence generation by metropolis-hastings sampling. In *Proceedings of the 33rd Conference on Artificial Intelligence*, 2019.
- C. A. Nicolaou, C. Kannas, and Erika Loizidou. Multi-objective optimization methods in de novo drug design. *Mini-reviews in Medicinal Chemistry*, 12(10):979–987, 2012.
- AkshatKumar Nigam, Pascal Friederich, Mario Krenn, and Alán Aspuru-Guzik. Augmenting genetic algorithms with deep neural networks for exploring the chemical space. In *Proceedings of the 8th International Conference on Learning Representations*, 2020.
- Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, and Hongming Chen. Molecular de-novo design through deep reinforcement learning. *Journal of Cheminformatics*, 9(1):48–48, 2017.
- W. Paszkowicz. Properties of a genetic algorithm equipped with a dynamic penalty function. *Computational Materials Science*, 45(1):77–83, 2009.
- Pavel G. Polishchuk, Timur I. Madzhidov, and Alexandre Varnek. Estimation of the size of drug-like chemical space based on gdb-17 data. *Journal of Computer-aided Molecular Design*, 27(8): 675–679, 2013.
- Mariya Popova, Olexandr Isayev, and Alexander E Tropsha. Deep reinforcement learning for de novo drug design. *Science Advances*, 4(7), 2018.
- Mariya Popova, Mykhailo Shvets, Junier Oliva, and Olexandr Isayev. Molecularrnn: Generating realistic molecular graphs with optimized properties. *arXiv: Learning*, 2019.
- David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010.

- Jeffrey S. Rosenthal. Optimal proposal distributions and adaptive mcmc. In *Handbook of Markov Chain Monte Carlo*, chapter 4. CRC Press, 2011.
- Daniel Schwalbe-Koda and Rafael Gómez-Bombarelli. Generative models for automatic chemical design. In *Machine Learning Meets Quantum Physics*. 2020.
- Ari Seff, Wenda Zhou, Farhan Damani, Abigail Doyle, and Ryan P. Adams. Discrete object generation with reversible inductive construction. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems, 2019*.
- Chence Shi, Minkai Xu, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang. Graphaf: a flow-based autoregressive model for molecular graph generation. In *Proceedings of the 8th International Conference on Learning Representations*, 2020.
- Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.
- Robin Winter, Floriane Montanari, Andreas Steffen, Hans Briem, Frank Noé, and Djork-Arné Clevert. Efficient multi-objective molecular optimization in a continuous latent space. *Chemical Science*, 10(34):8016–8024, 2019.
- Jiaxuan You, Bowen Liu, Zhitao Ying, Vijay S. Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems, 2018*.
- Chengxi Zang and Fei Wang. Moflow: An invertible flow model for generating molecular graphs. In *The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2020.
- Huangzhao Zhang, Hao Zhou, Ning Miao, and Lei Li. Generating fluent adversarial examples for natural languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, 2019.
- Maosen Zhang, Nan Jiang, Lei Li, and Yexiang Xue. Language generation via combinatorial constraint satisfaction: A tree search enhanced Monte-Carlo approach. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2020.

# Appendix

## A PROPERTY SCORES OF SAMPLED MOLECULES

The property score distributions of sampled  $N = 5000$  molecules of the GSK3 $\beta$ +JNK3+QED+SA setting are shown in Figure 5. The average of the metrics over the sampling path is shown in Figure 6.

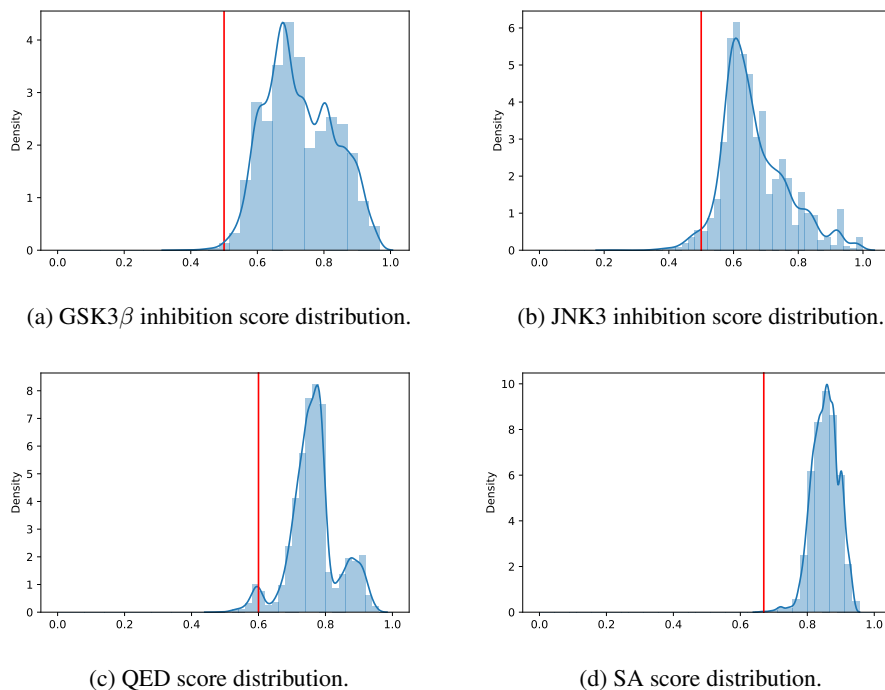


Figure 5: Property score distributions of sampled  $N = 5000$  molecules. The red lines are success thresholds.

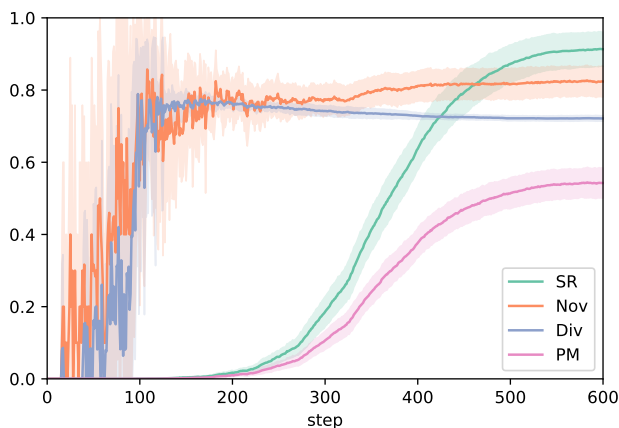


Figure 6: MARS sampling curves (average of 10 runs) for the GSK3 $\beta$ +JNK3+QED+SA setting. SR: success rate. Nov: novelty. Div: diversity. PM: product of the three metrics. Shaded area shows the standard deviations over 10 independent runs.

## B SINGLE OBJECTIVE GENERATION

To study whether our proposed method is capable of single-objective molecular generation, we also investigate how MARS performs on the drug-likeness (QED) and the penalized octanol-water partition coefficient (penalized logP) optimization. The experiment results are shown in Table 4. In the experiments, our approach can obtain the best performance on both QED and logP optimization. And especially, MARS outperforms previous methods significantly in the logP generation task.

Table 4: Comparison of different methods on single-objective molecular generation. Results of other baselines are taken from Shi et al. (2020) and Nigam et al. (2020).

Method	QED			Penalized logP		
	1st	2nd	3rd	1st	2nd	3rd
GCPN (You et al., 2018)	<b>0.948</b>	0.947	0.946	7.98	7.85	7.80
JT-VAE (Jin et al., 2018)	0.925	0.911	0.91	5.30	4.93	4.49
GraphAF (Shi et al., 2020)	<b>0.948</b>	<b>0.948</b>	0.947	12.23	11.29	11.05
GB-GA (Jensen, 2019)	/	/	/	15.76 ± 5.71	/	/
GA+D (Nigam et al., 2020)	/	/	/	20.72 ± 3.14	/	/
MARS	<b>0.948</b>	<b>0.948</b>	<b>0.948</b>	44.99	44.32	43.81

Moreover, from the results, we also can see how these two previously widely used metrics (Jin et al., 2018; You et al., 2018; Popova et al., 2019; Shi et al., 2020; Nigam et al., 2020) are questionable for both scientific study and practical use. Most of the generative methods (i.e., GCPN, JT-VAE, and GraphAF) can produce molecules with the highest possible QED score of 0.948, making the top QED score metric hard to distinguish different methods. As for logP optimization, heuristic search-based (i.e., GB-GA and GA+D) and sampling-based methods (i.e., MARS) can all easily beat generative models. This is because penalized logP score will prefer larger molecules that generative models can hardly produce. However, such large molecules are unrealistic for practical drug discovery, making the top penalized logP score metric problematic.

## C EXAMPLES OF SAMPLED MOLECULES

We also provide some examples of sampled molecules from the GSK3 $\beta$ +JNK3+QED+SA setting. The numbers under molecule graphs are GSK3 $\beta$ , JNK3, QED, and SA scores, respectively.

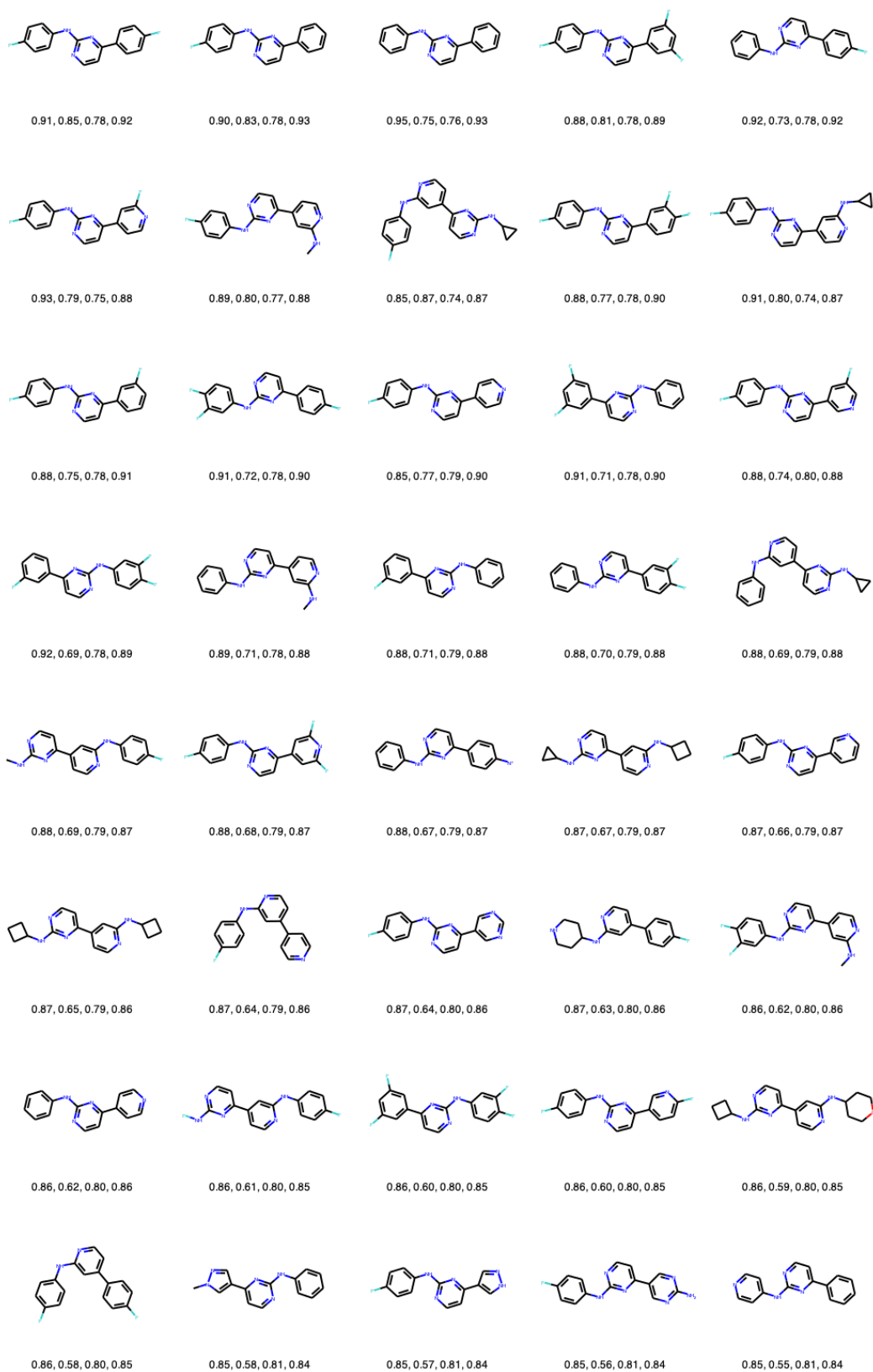
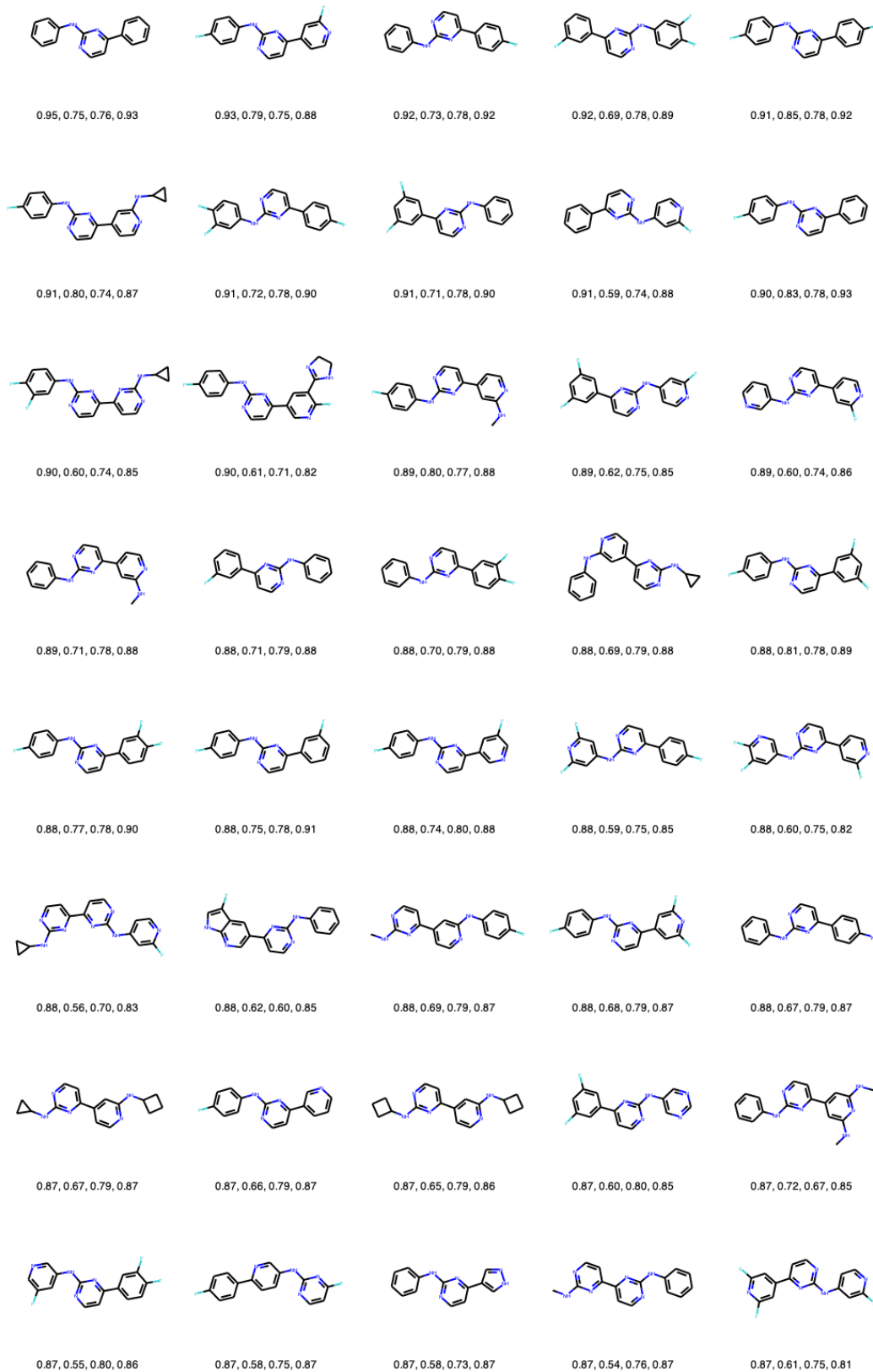


Figure 7: 40 sampled molecules with highest average property scores.

Figure 8: 40 sampled molecules with highest GSK3 $\beta$  scores.



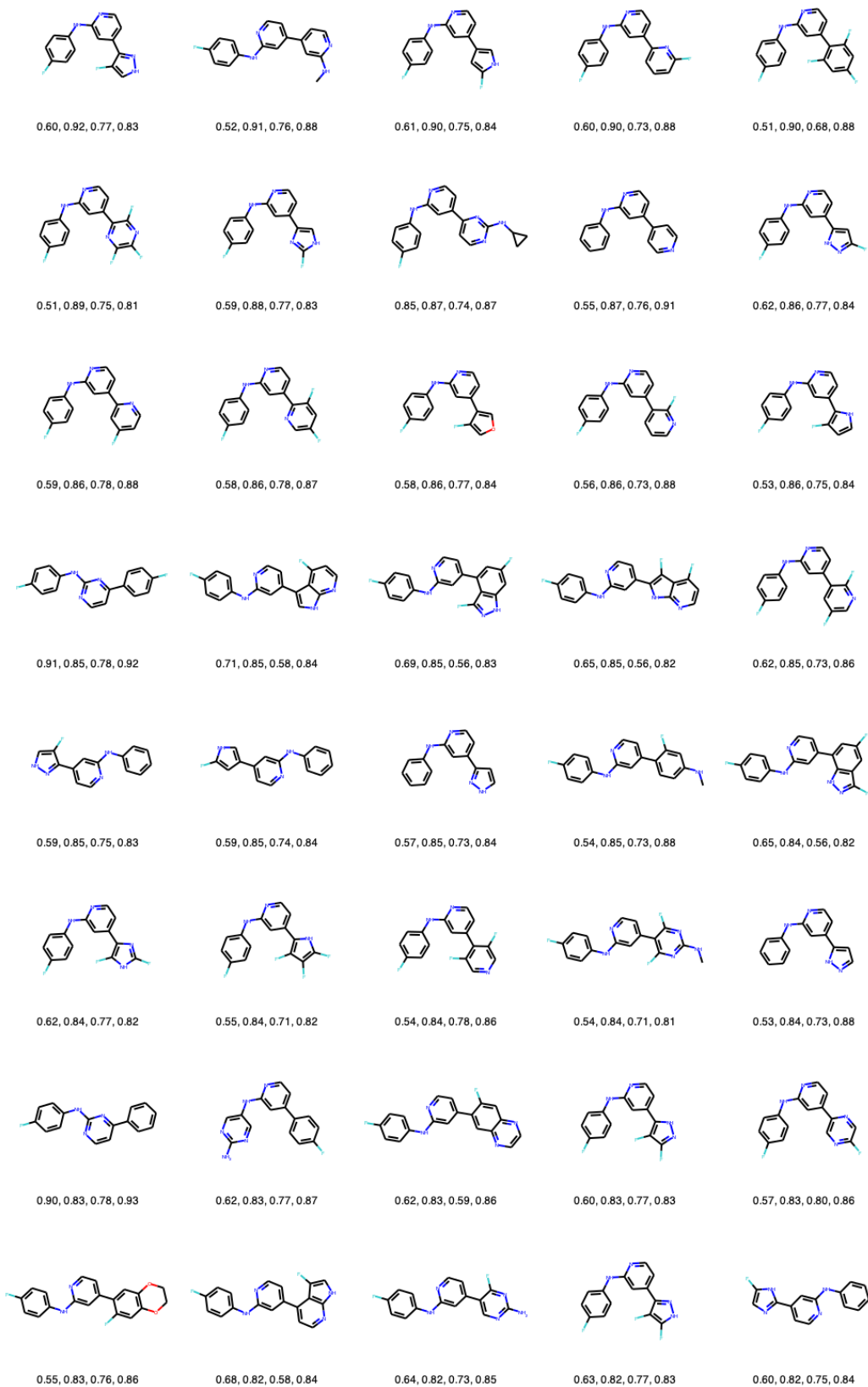


Figure 9: 40 sampled molecules with highest JNK3 scores.

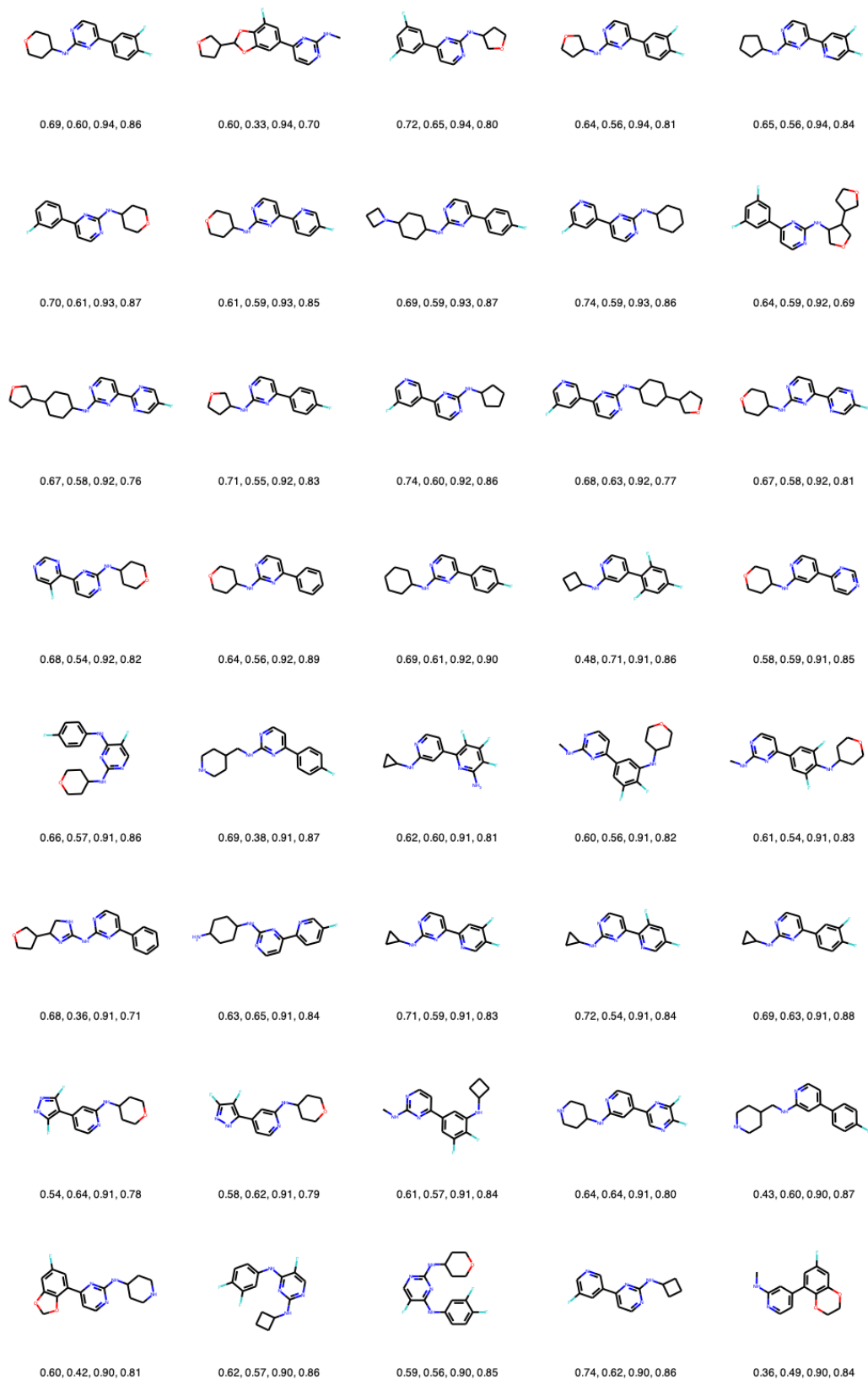


Figure 10: 40 sampled molecules with highest QED scores.



Figure 11: 40 sampled molecules with highest SA scores.