Learning to Navigate The Synthetically Accessible Chemical Space Using Reinforcement Learning

Sai Krishna Gottipati [*1] Boris Sattarov [*1] Sufeng Niu [2] Yashaswi Pathak [1 3] Haoran Wei [1 4] Shengchao Liu [5 6]
Karam J. Thomas [1] Simon Blackburn [6] Connor W. Coley [7] Jian Tang [8 6 9] Sarath Chandar [10 8 6]
Yoshua Bengio [5 11 8 6]

Abstract

Over the last decade, there has been significant progress in the field of machine learning for de novo drug design, particularly in generative modeling of novel chemical structures. However, current generative approaches exhibit a significant challenge: they do not ensure that the proposed molecular structures can be feasibly synthesized nor do they provide the synthesis routes of the proposed small molecules, thereby seriously limiting their practical applicability. In this work, we propose a novel reinforcement learning (RL) setup for de novo drug design: Policy Gradient for Forward Synthesis (PGFS), that addresses this challenge by embedding the concept of synthetic accessibility directly into the de novo drug design system. In this setup, the agent learns to navigate through the immense synthetically accessible chemical space by subjecting initial commercially available molecules to valid chemical reactions at every time step of the iterative virtual synthesis process. The proposed environment for drug discovery provides a highly challenging test-bed for RL algorithms owing to the large state space and high-dimensional continuous action space with hierarchical actions. PGFS achieves state-of-the-art performance in generating structures with high QED and clogP. Moreover, we validate PGFS in an in-silico proof-of-concept associated with three HIV targets. Finally, we describe how the end-to-end training conceptualized in this study represents an important paradigm in radically expanding the synthesizable chemical space and automating the drug discovery process.

1. Introduction

In the last decade, the role of machine learning and artificial intelligence techniques in chemical sciences and drug discovery has substantially increased (Schneider (2018); Butler et al. (2018); Goh et al. (2017)). Deep generative models such as GANs and VAEs have emerged as promising new techniques to design novel molecules with desirable properties (Sanchez-Lengeling & Aspuru-Guzik (2018); Assouel et al. (2018); Elton et al. (2019)). Generative models using either string-based (e.g., Segler et al. (2017)) or graph-based representations (e.g., Jin et al. (2018)) are able to output chemically valid molecules in a manner that can be biased towards properties like drug-likeness.

However, the majority of de novo drug design methodologies do not explicitly account for synthetic feasibility, and thus cannot ensure whether the generated molecules can be produced in the physical world. Synthetic complexity scores (Ertl & Schuffenhauer (2009); Coley et al. (2018b)) can be introduced into the scoring function to complement generative models. However, like any other data-driven predictive model, these heuristics are prone to exploitation by the generator, i.e, certain generated molecules with high accessibility scores will still be impossible or challenging to produce (Gao & Coley (2020)). Even though there is great work that has been done in the field of computer aided synthesis planning (Szymkuc et al. (2016); Segler et al. (2018); Coley et al. (2018a; 2019b)), relying on these programs creates a disjoint search pipeline that necessitates a separate algorithm for molecule generation and never guarantees that the generative model learns anything about synthesizability.

Directly embedding synthetic knowledge into de novo drug design would allow us to constrain the search to synthetically-accessible routes and theoretically guarantee

[*] Equal contribution [1] 99andBeyond, Montreal, Canada [2] Clemson University, South Carolina [3] Center for Computational Natural Sciences and Bioinformatics, IIIT Hyderabad, India [4] University of Delaware [5] University of Montreal, Montreal, Canada [6] Mila - Quebec AI Institute [7] Department of Chemical Engineering, Massachusets Institute of Technology, Massachusetts, USA [8] Canada CIFAR AI Chair [9] HEC Montréal [10] École Polytechnique de Montréal, Montreal, Canada [11] CIFAR Senior Fellow. Correspondence to: Sai Krishna <saikrishnagv1996@gmail.com>, Boris Sattarov <brois475@gmail.com>.

Proceedings of the 37th International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

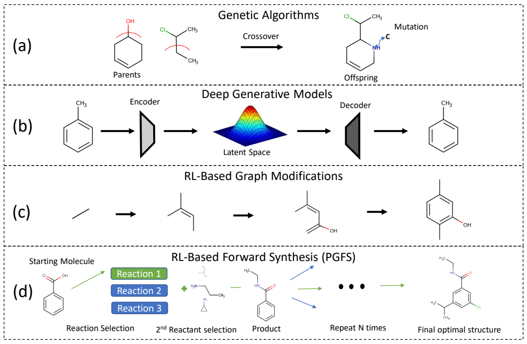


Figure 1. Illustrative comparison of de novo drug design methodologies including: (a) genetic algorithms (Brown et al. (2004); Jensen (2019)); (b) deep generative models (Simonovsky & Komodakis (2018); Gómez-Bombarelli et al. (2018); Winter et al. (2019); Jin et al. (2018); Popova et al. (2018); Olivecrona et al. (2017)); (c) RL-based graph modifications (You et al. (2018a); Zhou et al. (2018)); and (d) RL-based forward synthesis as proposed in our methodology Policy Gradient for Forward Synthesis (PGFS).

that any molecule proposed by the algorithm can be easily produced. To accomplish this, we present a forward synthesis model powered by reinforcement learning (RL) entitled Policy Gradient for Forward Synthesis (PGFS) that treats the generation of a molecular structure as a sequential decision process of selecting reactant molecules and reaction transformations in a linear synthetic sequence. The agent learns to select the best set of reactants and reactions to maximize the task-specific desired properties of the product molecule, i.e., where the choice of reactants is considered an action, and a product molecule is a state of the system obtained through a trajectory composed of the chosen chemical reactions. The primary contribution of this work is the development of a RL framework able to cope with the vast discrete action space of multi-step virtual chemical synthesis and bias molecular generation towards chemical structures that maximize a black-box objective function, generating a full synthetic route in the process. We define the problem of de novo drug design via forward synthesis as a Markov decision process in chemical reaction space, and we propose to search in a continuous action space using a relevant feature space for reactants rather than a discrete space to facilitate the learning of the agent. Training is guided by rewards which correspond to the predicted properties of the resulting molecule relative to the desired properties. We show that our algorithm achieves state-of-the-art performance on standard metrics like quantitative estimate of drug-likeness (QED) (Bickerton et al. (2012)) and penalized octanol-water partition as defined by You et al. (2018b). Furthermore, as a proof-of-concept, our algorithm generated molecules with higher predicted activity against three HIV-related biological targets relative to existing benchmarks. The HIV targets activity datasets used, predictive QSAR models and prediction scripts can be found at this url: `https://github.com/99andBeyond/Apollo1060`

2. Related Work

To highlight the improvements we are proposing in this work, we focus the discussion on de novo drug design methodologies that can perform single- and multi-objective optimization of chemical structures.

2.1. Genetic Algorithms

Genetic algorithms (GA) have been used for many decades to generate and optimize novel chemical structures. The majority of published GA approaches (Brown et al. (2004); Jensen (2019)) use graph-based representations of the molecule and apply specific graph sub-fragments crossover operations to produce offsprings followed by mutation operations in the form of random atom, fragment and bond type replacements. More recently, string-based representations of molecules were also proposed in the GA optimization setting (Krenn et al. (2019); Nigam et al. (2019)). Existing implementations of GA for de novo generation can only account for synthetic feasibility through the introduction of a heuristic scoring functions (Ertl & Schuffenhauer (2009); Coley et al. (2018b)) as part of the reward function. As a result, they need a separate model for retrosynthesis or manual evaluation by an expert upon identifying a structure with desired properties.

2.2. Deep Generative Models

Many recent studies highlight applications of deep generative systems in multi-objective optimization of chemical structures (Gómez-Bombarelli et al. (2018); Winter et al. (2019)). Other recent publications describe improvements in learning by utilizing RL (Olivecrona et al. (2017); Popova et al. (2018); Guimaraes et al. (2017)). While these approaches have provided valuable techniques for optimizing various types of molecular properties in single- and multi-objective settings, they exhibit the same challenges in synthetic feasibility as genetic algorithms.

2.3. RL-Based Graph Modification Models

You et al. (2018b) and Zhou et al. (2018) recently proposed reinforcement learning based algorithms to iteratively modify a molecule by adding and removing atoms, bonds or molecular subgraphs. In such setups, the constructed molecule M_t, represents the state at time step t. The state at time step 0 can be a single atom like carbon or it can be completely null. The agent is trained to pick actions that would optimize the properties of the generated molecules. While these methods have achieved promising results, they do not guarantee synthetic feasibility.

2.4. Forward Synthesis Models

The generation of molecules using forward synthesis is the most straightforward way to deal with the problem of synthetic accessibility. Generalized reaction transformations define how one molecular subgraph can be produced from another and can be encoded by expert chemists (Hartenfeller et al. (2012); Szymkuc et al. (2016)) or algorithmically extracted from reaction data (Law et al. (2009)). Libraries of these "templates" can be used to enumerate hypothetical product molecules accessible from libraries of available starting materials. In fact, de novo drug design via forward synthesis isn't a new concept, and has been used for decades to generate chemical libraries for virtual screening (Walters (2018)). Templates can be used in a goal-directed optimization setting without relying on complete enumeration. Vinkers et al. (2003) describe an iterative evolutionary optimization approach called SYNOPSIS to produce chemical structures with optimal properties using reaction-based transformations. Patel et al. (2009) explored the enumeration and optimization of structures by taking advantage of the reaction vectors concept. More recently, many approaches focused on reaction-based enumeration of analogs of known drugs and lead compounds have been proposed (Hartenfeller et al. (2012); Button et al. (2019)). Although promising results were reported when using a reaction-based enumeration approach that was followed by an active learning module (Konze et al. (2019)), mere enumeration severely limits the capacity of the model to explore the chemical space efficiently.

Recently, Bradshaw et al. (2019) and Korovina et al. (2019) have proposed approaches to de novo drug design that use reaction prediction algorithms to constrain the search to synthetically-accessible structures. Bradshaw et al. (2019) use a variational auto-encoder to embed reactant structures and optimize the molecular properties of the resulting product from the *single-step* reaction by biasing reactant selection. Korovina et al. (2019) propose an algorithmically simpler approach, whereby random selection of reactants and conditions are used to stochastically generate candidate structures, and then subject the structures to property evaluation. This workflow produces molecules through multi-step chemical synthesis, but the selection of reactants cannot be biased towards the optimization objective. We combine the unique strengths of both frameworks (biased generation and multi-step capabilities) in our approach; in doing so, we make use of a novel RL framework.

2.5. Benchmarking De Novo Drug Design

It is difficult to properly evaluate approaches for de novo drug design without conducting the actual synthesis of the proposed compounds and evaluating their properties in laboratory experiments. Yet, several simple benchmarks have been adopted in recent publications. Metrics like the Frechenet ChemNet distance (Preuer et al. (2018)) aim to measure the similarity of the distributions of the generated structures relative to the training set. Objective-directed benchmarks evaluate the ability to conduct efficient single- and multi-objective optimization for the proposed structures. The most widely used objective functions are QED (Bickerton et al. (2012)), a quantitative estimate of drug-likeness, and penalized clogP as defined by You et al. (2018b), an estimate of the octanol-water partition coefficient that penalizes large aliphatic cycles and molecules with large synthetic accessibility scores (Ertl & Schuffenhauer (2009)). While these metrics enable the comparison of systems with respect to their ability to optimize simple reward functions associated with the proposed structures, they bear little resemblance to what would be used in a real drug discovery project. Recently, two efforts in creating benchmarking platforms have been described in the corresponding publications: MOSES (Polykovskiy et al. (2018)) and GuacaMol (Brown et al. (2019)). While MOSES focuses on the distribution of properties of the generated structures, GuacaMol aims to establish a list of goal-directed drug de novo design benchmarks based on the similarity to a particular compound, compound rediscovery and search for active compounds containing different core structures (scaffold hopping). In a recent review describing the current state of the field (Coley et al. (2019a)) of autonomous discovery, the authors state that the community needs to focus on proposing benchmarks that will better incorporate the complexity of the real-world drug discovery process such as ligand and structure based modeling.

3. Methods

3.1. Reinforcement Learning

To explore the large chemical space efficiently and maintain the ability to generate diverse compounds, we propose to consider a molecule as a sequence of unimolecular or bimolecular reactions applied to an initial molecule. PGFS learns to select the best set of commercially available reactants and reaction templates that maximize the rewards associated with the properties of the product molecule. This guarantees that the only molecules being considered are synthesizable and also provides the recipe for synthesis. The state of the system at each step corresponds to a product molecule and the rewards are computed according to the properties of the product. Furthermore, our method decomposes actions of synthetic steps in two sub-actions. A reaction template is first selected and is followed by the selection of a reactant compatible with it. This hierarchical decomposition considerably reduces the size of the action space in each of the time steps in contrast to simultaneously picking a reactant and reaction type.

However, this formulation still poses challenges for current state-of-the-art RL algorithms like PPO (Schulman et al. (2017)) and ACKTR (Wu et al. (2017)) owing to the large action space. In fact, there are tens of thousands of possible reactants for each given molecule and reaction template. As a result, we propose to adapt algorithms corresponding to continuous action spaces and map continuous embeddings to discrete molecular structures by looking up the nearest molecules in this representation space via a k-nearest neighbor (k-NN) algorithm. Deterministic policy gradient (Silver et al. (2014)) is one of the popular RL algorithms for continous action space. Deep deterministic policy gradient (DDPG) (Lillicrap et al. (2015)), Distributed distributional DDPG (D4PG) (Barth-Maron et al. (2018)) and Twin delayed DDPG (TD3) (Fujimoto et al. (2018)) constitute the consequent improvements done over DPG. Soft actor critic (SAC, Haarnoja et al. (2018)) also deals with continuous action spaces with entropy regularization. In this work, we leverage a TD3 algorithm along with the k-NN approach from Dulac-Arnold et al. (2015). There are three key differences with this work: (1) our actor module includes two learnable networks (instead of just one) to compute two levels of actions; (2) we do not use a critic network in the forward propagation, and include the k-NN computation as part of the environment. Thus, the continuous output of the actor module reflects the true actions–not a proto-action to be discretized to obtain the actual action; and (3) we leverage the TD3 algorithm which has been shown to be better than DPG (used in Dulac-Arnold et al. (2015)) on several RL tasks.

3.2. Overview

The pipeline is setup in such a way that at every time step t, a reactant $R_t^{(2)}$ is selected to react with the existing molecule $R_t^{(1)}$ to yield the product $R_{t+1}^{(1)}$ which is the molecule for the next time step. $R_t^{(1)}$ is considered as the current state s_t and our agent chooses an action a_t that is further used in computing $R_t^{(2)}$. The product $R_{t+1}^{(1)}$ (which is considered as the next state s_{t+1}) is determined by the environment based on the two reactants ($R_t^{(1)}$ and $R_t^{(2)}$). At the very initial time step, we randomly sample the initial molecule $R_0^{(1)}$ from the list of all commercially available reactants. To overcome the limitation of large discrete action space where there are over a hundred thousand possible second reactants, we introduce an intermediate action which reduces the space of reactants considered by choosing a reaction template. Reaction templates, encoded in the SMARTS (James et al. (2000)) language, define allowable chemical transformations according to subgraph matching rules. They can be applied deterministically to sets of reactant molecules to propose hypothetical product molecules using cheminformatics tools like RDKit (Landrum (2016)). One of the reactants is the state s_t while the other reactant is later selected. Since the required substructure of $R_t^{(2)}$ that can participate in the reaction and of the state s_t is determined by the choice of the reaction template, the action space comprising the space of all $R^{(2)}$s becomes constrained to those reactants which contain this particular substructure. We also enforce the additional constraint of having this substructure present only once in the structure. If multiple products are still possible the first product returned by RDKit's RunReactants function is selected. Even with the previous constraints, there can be tens of thousands of reactants at each step, which represents a challenge for traditional RL algorithms. Thus, we formulate a novel Markov Decision Process (MDP) involving a continuous action space.

The agent comprises three learnable networks f, π and Q. In terms of the actor-critic framework, our actor module Π comprises f and π networks and the critic is composed of the Q network that estimates the Q-value of the state-action pair. At any time step t, the input to the actor module is the state s_t ($R_t^{(1)}$) and the output is the action a_t which is a tensor defined in the feature representation space of all initial reactants $R^{(2)}$. The f network predicts the best reaction template T_t given the current state s_t ($R_t^{(1)}$). Using the best reaction template T_t and ($R_t^{(1)}$) as inputs, the π network computes the action a_t. The environment takes the state s_t, best reaction template T_t, and action a_t as inputs and computes the reward r_t, next state s_{t+1} and a boolean to determine whether the episode has ended. It first chooses k reactants from the set $R^{(2)}$ corresponding to the k-closest embeddings to the action a using the k nearest neighbours technique in which we pre-compute feature representations for all reactants. Each of these k actions are then passed through a reaction predictor to obtain the corresponding k products. The rewards associated with the products are computed using a scoring function. The reward and product corresponding to the maximum reward are returned. The state s_t, best template T_t, action a_t, next state s_{t+1}, reward r_t are stored in the replay memory buffer. The episode terminates when either the maximum number of reaction steps is reached or when the next state has no valid templates. In our experiments, we have 15 unimolecular and 82 bimolecular reaction templates. The unimolecular templates do not require selection of an $R^{(2)}$, and hence for such cases we directly obtain the product using $R_t^{(1)}$ and the selected T_t.

During initial phases of the training, it is important to note that the template chosen by the f network might be invalid. To overcome this issue and to ensure the gradient propagation through the f network, we first multiply the template T with the template mask T_{mask} and then use Gumbel softmax

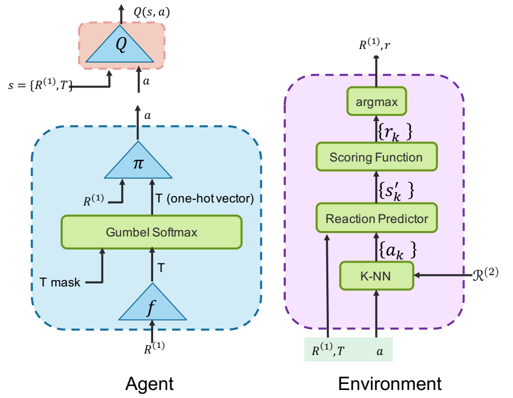


Figure 2. PGFS environment and agent. The environment takes in the state s ($R^{(1)}$), the reaction template T, the action a (a tensor in the space of feature representations of reactants) and outputs the next state s' ($R^{(1)}$ for next time step) and reward r. $\{a_k\}$ is the set of top-k reactants closest to the action a, obtained using the k-nearest neighbours algorithm. The reaction predictor computes k products (next states) corresponding to the k reactants when they react with $R^{(1)}$. The scoring function computes k rewards corresponding to the k next states. Finally, the next state corresponding to the maximum reward is chosen. The agent is composed of actor and critic modules. The actor predicts the action a given the state input $R^{(1)}$ and the critic evaluates this action.

to obtain the best template:

$$T = T \odot T_{mask}$$

$$T = GumbelSoftmax(T, \tau)$$

where, τ is the temperature parameter that is decayed at every time step by multiplying with a decay parameter until the τ reaches a minimum threshold of 0.1.

3.2.1. Training Paradigm

The learning agent can be trained using any policy gradient algorithm applicable for continuous action spaces. Thus, we call our algorithm "Policy Gradient for Forward Synthesis (PGFS)". DDPG (Lillicrap et al. (2015)) is one of the first deep RL algorithms for continuous action spaces. After sampling a random minibatch of N transitions from the buffer, the actor and critic modules are updated as follows: The critic (Q-network) is updated using the one-step TD update rule as:

$$y_i = r_i + \gamma Q'(s_{i+1}, \Pi'(s_{i+1}))$$

where, Q' and Π' are the target critic and actor networks respectively, i.e, they are a copy of the original networks but they do not update their parameters during gradient updates. y_i is the one-step TD target, r_i is the immediate reward and s_i constitutes the state at the time step t. s_{i+1} forms the state at next time step. The critic loss is then:

$$L = \frac{1}{N} \sum_i (y_i - Q(s_i, a_i))^2$$

and the parameters of the Q network are updated via back propagation of the critic loss. The goal of the actor module is to maximize the overall return (weighted average of future rewards) achieved over the given initial distribution of states by following the actions determined by the actor module. The Q network can be seen as an approximation to this overall return. Thus, the actor should predict actions that maximize the $Q(s, a)$ values predicted by Q network i.e, $\max Q(s, \Pi(s))$, or $\min -Q(s, \Pi(s))$. Thus, $-Q(s, \Pi(s))$ constitutes the actor loss. Consequently, the parameters of the actor module (of f and π networks) are updated towards reducing that loss.

However, the convergence of returns observed is slow because of the reasons highlighted by Fujimoto et al. (2018). Accordingly, we use the approach from (Fujimoto et al., 2018) for faster training.

Firstly, we smooth the target policy (akin to regularization strategy) by adding a small amount of clipped random noises to the action.

$$\tilde{a} = a + \epsilon; \quad \epsilon \sim \text{clip}(N(0, \bar{\sigma}), -c, c)$$

We use a double Q-learning strategy comprising two critics, but only consider the minimum of two critics for computing the TD target:

$$y = r + \gamma \min_{i=1,2} Q_i(s', \Pi(s'))$$

Further, we make delayed updates (typically, once every two critic updates) to the actor module and target networks.

To speed up the convergence of the f network, we also minimize the cross entropy between the output of the f network and the corresponding template T obtained for the reactant $R^{(1)}$.

Firstly, we smooth the target policy (akin to regularization strategy) by adding a small amount of clipped random noises to the action.

$$\tilde{a} = a + \epsilon; \quad \epsilon \sim \text{clip}(N(0, \bar{\sigma}), -c, c)$$

We use a double Q-learning strategy comprising two critics, but only consider the minimum of two critics for computing the TD target:

$$y = r + \gamma \min_{i=1,2} Q_i(s', \Pi(s'))$$

Further, we make delayed updates (typically, once every two critic updates) to the actor module and target networks.

To speed up the convergence of the f network, we also minimize the cross entropy between the output of the f network and the corresponding template T obtained for the reactant $R^{(1)}$.

Algorithm 1 PGFS

```
procedure ACTOR($R^{(1)}$)
    $T \leftarrow f(R^{(1)})$
    $T \leftarrow T \odot T_{mask}$
    $T \leftarrow GumbelSoftmax(T, \tau)$
    $a \leftarrow \pi(R^{(1)}, T)$
    return $T, a$
procedure CRITIC($R^{(1)}$, $T$, $a$)
    return $Q(R^{(1)}, T, a)$
procedure ENV.STEP($R^{(1)}, T, a$)
    $\mathcal{R}^{(2)} \leftarrow$ GetValidReactants($T$)
    $\mathcal{A} \leftarrow$ kNN($a, \mathcal{R}^{(2)}$)
    $\mathcal{R}^{(1)}_{t+1} \leftarrow$ ForwardReaction($R^{(1)}, T, \mathcal{A}$)
    $\mathcal{R}ewards \leftarrow$ ScoringFunction($\mathcal{R}^{(1)}_{t+1}$)
    $r_t, R^{(1)}_{t+1}, done \leftarrow \arg\max \mathcal{R}ewards$
    return $R^{(1)}_{t+1}, r_t$, done
procedure BACKWARD(buffer minibatch)
    $T_{i+1}, a_{i+1} \leftarrow$ Actor-target($R^{(1)}_{i+1}$)
    $y_i \leftarrow r_i + \gamma \min_{j=1,2}$ Critic-
  target($\{R^{(1)}_{i+1}, T_{i+1}\}, a_{i+1}$)
    min $L(\theta^Q) = \frac{1}{N}\sum_i |y_i -$ Critic($\{R^{(1)}_i, T_i\}, a_i$)$|^2$
    $\min L(\theta^{f,\pi}) = -\sum_i Critic(R^{(1)}_i, Actor(R^{(1)}_i))$
    $\min L(\theta^f) = -\sum_i (T^{(1)}_i, log(f(R^{(1)}_i)))$
procedure MAIN($f$, $\pi$, $Q$)
    for episode = 1, M do
        sample $R^{(1)}_0$
        for t = 0, N do
            $T_t, a_t \leftarrow$ Actor($R^{(1)}_t$)
            $R^{(1)}_{t+1}, r_t$, done $\leftarrow$ env.step($R^{(1)}_t, T_t, a_t$)
            store ($R^{(1)}_t, T_t, a_t, R^{(1)}_{t+1}, r_t$, done) in buffer
            sample a random minibatch from buffer
            Backward(minibatch)
  =0
```

4. Experiments

4.1. Predictive Modeling

To test the applicability of PGFS in an in-silico proof-of-concept for de novo drug design, we develop predictive models against three biological targets related to the human immunodeficiency virus (HIV) - as scoring functions. The biological activity data available in the public domain allowed us to develop ligand-based machine learning models using the concept of quantitative structure-activity relationship modeling (QSAR).

HIV Targets i) The first target in this study, C-C chemokine receptor type 5 (CCR5), is a receptor located on the surface of the host immune cells. Along with C-X-C chemokine receptor type 4 (CXCR4), this receptor is used by HIV to recognize target cells. Hence, antagonists of this receptor allows HIV entry inhibition (Arts & Hazuda (2012)).

ii) The second target is HIV integrase that catalyzes HIV viral DNA processing and strand transfer. Inhibitors of that enzyme target the strand transfer reaction, thus allowing for HIV integration inhibition.

iii) The last selected target is HIV Reverse transcriptase (HIV-RT) which was the first enzyme used as biological target in antiretroviral drug discovery. It is an enzyme with multiple functions that are necessary to convert the single strand of the viral RNA to a double stranded DNA.

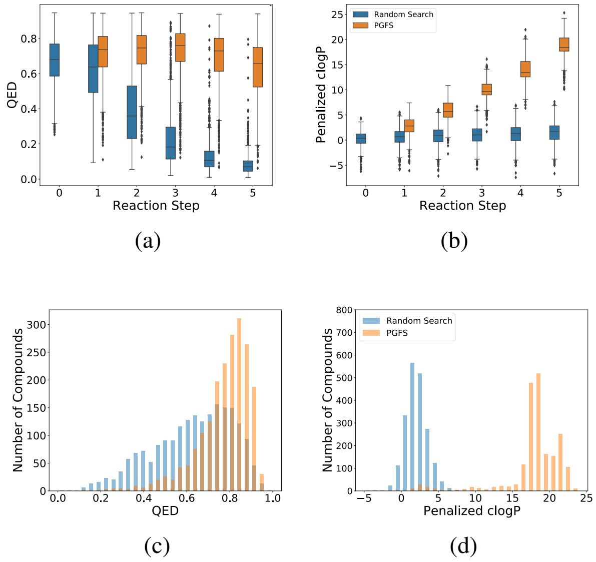
Quantitative Structure Activity Relationships The goal of QSAR studies is to discover functional relationship between the structure of the chemical compound and its activity relative to a biological target of interest (Cherkasov et al. (2014)). Widely accepted guidelines for building QSAR models were developed in the related publications (Tropsha (2010); Cherkasov et al. (2014); Muratov et al. (2020)) describing training data curation, testing models performance, usage of the Applicability Domain (AD) and more. We trained our QSAR models to predict the compounds' pIC_{50} ($-\log_{10}IC_{50}$ where the IC_{50} is the molar concentration of a compound that produces a half-maximum inhibitory response) values associated with three HIV-related targets reported in the ChEMBL database. The description of the data curation, QSAR training procedures, definition of AD and predictive performance of the models developed in this study can be found in Section-3 of the Appendix.

4.2. Data and Representations

ECFP4-Like Morgan Fingerprints We utilize Morgan circular molecular fingerprint bit vector of size 1024 and radius 2 as implemented in RDKit (Landrum (2016)) with default invariants that use connectivity information similar to those used for the ECFP fingerprints (Rogers & Hahn (2010)). Generally, Morgan fingerprints utilize the graph topology, and thus can be viewed as a learning-free graph representation. Moreover, some recent studies (Liu et al. (2019)) demonstrate its performance is competitive to the state-of-the-art Graph Neural Network.

MACCS Public Keys We leveraged 166 public MACCS keys as implemented in RDKit. MACCS keys constitute a very simple binary feature vector where each bin corresponds to the presence (1) or to the absence (0) of the pre-defined molecular sub-fragment.

Molecular Descriptors Set (MolDSet) The set of normalized molecular continuous descriptors are selected from the 199 descriptors available in RDKit (Landrum



(a) (b)

(c) (d)

Figure 3. Performance comparison of Random Search (RS) vs. PGFS using the validation set of initial reactants (R1s) and corresponding rewards. (a) and (b): box plots of the QED and penalized clogP scores per step of the iterative five-step virtual synthesis. The first step (Reaction Step = 0) in each box plot shows the scores of the initial reactants (R1s). (c) and (d): distributions of the maximum QED and penalized clogP scores over five-step iterations. A few outliers with penalized clogP lower than -10 with both methods were clipped out when plotting (b) and (d).

(2016)).The set consists of 35 descriptors that were picked as most important during the feature selection process during HIV-related QSAR modeling presented in this paper. The resulting set of 35 features consists of descriptors such as maximum, minimum and other e-state indices (Kier et al. (1999)), molecular weight, Balaban's J index (Balaban (1982)) among others. The full list of descriptors used in this set is reported in Section-1 of the Appendix.

We have experimented with several feature representations and observed that MolDSet works best as input features to the k-NN module (and thus as the output of the actor module) and ECFP works best as input to the f, π and Q networks. The results reported in this paper use only these two features. Further analysis is provided in Section-1 of the Appendix.

Reaction Templates And Reactants The structures of reactants used in this study originate from the Enamine Building Block catalogue[1] Global Stock. Only $150,560$ unique (with stereo) building blocks that matched at least one reaction template as a first or second reactants were used in this study. The full list of SMILES of the building blocks can be found in the github repository of this work.

The set of reaction templates used in this study was taken from Button et al. (2019). Several templates were additionally manually curated to resolve occasional errors such as broken aromaticity and stereochemistry problems upon using them with RDKit *RunReactants* function (version 2019.03.1). We note that only stereocenters specified in the initial reactants are kept in the products and stereocenters that would be formed during the reaction are left without specification. This is one of the limitations of using reaction templates that cannot accurately predict stereoselectivity. Since the reaction template selection is based on the reactant that will be used as the first one in the reaction, the 49 bimolecular templates were transformed into 98. For example, the "Michael addition" template also consists of the "Michael addition as R2". The 15 unimolecular templates are also used. We additionally filter out reaction templates that have fewer than 50 available second reactants for them, resulting in a final set of 97 templates. Additional statistics and examples of reaction templates are provided in Section-2 of the Appendix.

Datasets For QSAR Modeling The datasets for all three HIV targets were downloaded from ChEMBL ((Gaulton et al., 2017)) corresponding to the following target IDs: HIV-RT - CheMBL247, HIV-Integrase - CheMBL3471, CCR5 - CheMBL274. The full datasets used for QSAR modeling are provided in the github repository. The data curation procedure is described in Section-3 of the Appendix.

4.3. Experimental Settings

Model Setup Hyper parameter tuning was performed and the following set of parameters were used in all the experiments reported in this paper. The f network uses four fully connected layers with 256, 128, 128 neurons in the hidden layers. The π network uses four fully connected layers with 256, 256, 167 neurons in the hidden layers. All the hidden layers use ReLU activation whereas the final layer uses tanh activation. Similarly, the Q network also uses four fully connected layers with 256, 64, 16 neurons in the hidden layers, with ReLU activation for all the hidden layers and linear activation for the final layer. We use the Adam optimizer to train all the networks with a learning rate of 1e-4 for the f and π networks and 3e-4 for the Q network. Further, we used a discount factor $\gamma = 0.99$, mini batch size = 32, and soft update weight for target networks, $\tau = 0.005$. We have only used $k = 1$ (in the k-NN module) during both the training and inference phases of our algorithm for fair comparison.

Baseline Setup The specific baseline in this study, Random Search (RS) starts with a random initial reactant ($R^{(1)}$) followed by the selection of a random reaction template T, and then the random selection of a compatible reactant $R^{(2)}$. The product of the reaction is used as the $R^{(1)}$ in the

[1] https://enamine.net/building-blocks

next reaction. This process is repeated until the maximum allowed number of synthesis steps is reached or until the product doesn't have any reactive centers left. In this study, we define the maximum number of synthesis steps allowed in an episode to be five. The random search continues until the stop criterion such as search time or number of reactions is reached. The total number of allowed reaction steps used during random search to produce results in Table 1 and Table 2 is 400,000.

4.4. Results and Analysis

Table 1. Performance comparison of the maximum achieved value with different scoring functions. The reported HIV-related QSAR-based scoring functions RT, INT and CCR5 correspond to structures inside of the AD of the predictive ensemble. If a structure with maximum value is outside of the AD, its value is reported in brackets. The compounds with the highest scores are presented in Appendix Section 2. The QED and penalized clogP values for JT-VAE, GCPN and MSO are taken from Jin et al. (2018), You et al. (2018a), Winter et al. (2019) respectively. The experiments performed to evaluate the models on HIV rewards are detailed in the appendix. Values corresponding to the initial set of building blocks are reported as ENAMINEBB.

Method	QED	clogP	RT	INT	CCR5
ENAMINEBB	**0.948**	5.51	7.49	6.71	8.63
RS	**0.948**	8.86	7.65	7.25	8.79 (8.86)
GCPN	**0.948**	7.98	7.42(7.45)	6.45	8.20(8.62)
JT-VAE	0.925	5.30	7.58	7.25	8.15 (8.23)
MSO	**0.948**	26.10	7.76	7.28	8.68 (8.77)
PGFS	**0.948**	**27.22**	**7.89**	**7.55**	**9.05**

4.4.1. Baseline comparison

PGFS performance on QED and penalized clogP rewards vs. Random Search(RS) - The validation set constitutes randomly chosen 2,000 $R^{(1)}$s initial reactants from the set of 150,560 available reactants. First, we carry out random search (RS) by randomly choosing reaction templates and second reactants (for bimolecular reactions) at every time step. Then, we use the trained models from PGFS (trained on QED for 200,000 time steps and on penalized clogP for 390,000 time steps) in the inference mode and calculate QED and penalized clogP of the products, using the validation set. We observe that our algorithm performs significantly better than the random baseline. We can observe a clear distribution shift of each score given the same initial compounds which confirms that the training was successful.

PGFS performance on HIV rewards vs. Random Search(RS) Next, we implement both these algorithms on HIV rewards and make a similar observation from Figure 4 that the rewards associated with the structures obtained using our method (PGFS) are substantially better than the RS method. Furthermore, we filter out compounds that do not satisfy the AD criteria of the QSAR model from both sets and still clearly observe the distribution shift towards high scoring compounds in case of PGFS using CCR5 reward in Figure 4(c). Similar shifts can be observed when using HIV-RT and HIV-Int rewards (Appendix Section 2, Figure 8). PGFS was trained on HIV-CCR5 for 310,000 time steps, HIV-INT for 120,000 time steps, HIV-RT for 210,000 time steps. All these models were pre-trained on the QED task for 100,000 time steps.

4.4.2. Quantitative performance benchmark

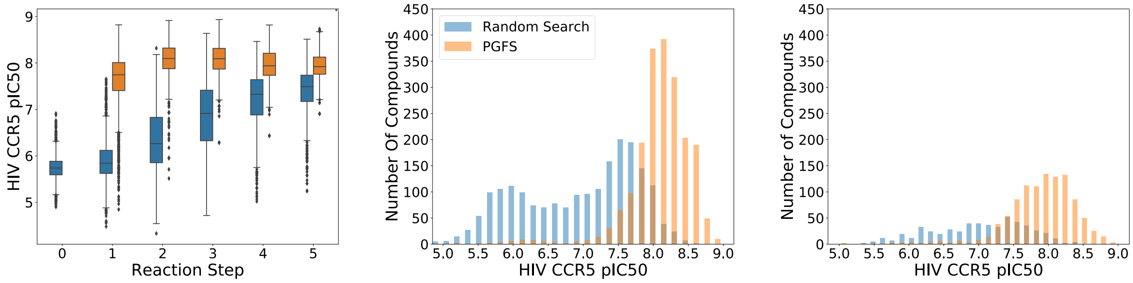
Table 1 compares our proposed model performance against various models on different scoring functions (Winter et al. (2019); You et al. (2018a); Jin et al. (2018)). Our proposed framework has produced compounds with the highest maximum scores compared to all other approaches on every defined task. PGFS achieved a maximum QED score reported in the de novo generative design studies. However, although our system cannot just return initial building block without any reactions applied to it, we can see that a set of initial building blocks (ENAMINEBB) already contains the compounds with the highest QED of 0.948. Random search was also successful in producing a maximum QED scoring compound. We also notice a significantly higher maximum penalized clogP value compared to the existing approaches, especially GCPN and JT-VAE. This is due to the fact that molecular size correlates with the heuristic penalized clogP score if the molecule contains hydrophobic moieties and that our method does not have restrictions (besides number of reaction steps) associated with the size of the produced structures in contrast to other methods; achievable values of the penalized clogP score strongly depend on the number of steps the reaction-based system is allowed to take.

Thus, QED and penalized clogP scores are insufficient to compare approaches designed to be used in real drug discovery setting. However, Figure 3 clearly demonstrates that PGFS training procedure was successful in biasing structures of virtually synthesised compounds towards high values of these scoring functions. In the proof of concept where the task is defined as a single-objective maximization of the predicted half maximal inhibitory concentration (pIC50) of the HIV-related targets, PGFS achieves the highest scores when compared to de novo drug design methods and random search in the maximum reward achieved (Table - 1) and mean of the top-100 highest rewards (Table - 2) comparisons, given the settings of this study.

Proof-Of-Concept Figure 5 demonstrates one of the proposed compounds with the highest predicted inhibitory activity (pIC50) against the CCR5 HIV target. As recommended by Walters & Murcko (2020), we also provide side by side comparison of the proposed structure with the most similar one in the training set utilized to build the QSAR model in the Appendix Section-2

Table 2. Mean $\pm 1 std$ of the top-100 produced molecules with highest predicted HIV scores for every method used and Enamine's building blocks. Only unique compounds were used after the stereo information was stripped to calculate the values presented in this table. *GCPN and MSO runs only produced 90 and 28 compounds inside the Applicability Domain (AD) of the CCR5 QSAR model, respectively.

	NO AD			AD		
Scoring	RT	INT	CCR5	RT	INT	CCR5
ENAMINEBB	6.87 ± 0.11	6.32 ± 0.12	7.10 ± 0.27	6.87 ± 0.11	6.32 ± 0.12	6.89 ± 0.32
RS	7.39 ± 0.10	6.87 ± 0.13	8.65 ± 0.06	7.31 ± 0.11	6.87 ± 0.13	8.56 ± 0.08
GCPN	7.07 ± 0.10	6.18 ± 0.09	7.99 ± 0.12	6.90 ± 0.13	6.16 ± 0.09	$6.95^* \pm 0.05$
JT-VAE	7.20 ± 0.12	6.75 ± 0.14	7.60 ± 0.16	7.20 ± 0.12	6.75 ± 0.14	7.44 ± 0.17
MSO	7.46 ± 0.12	6.85 ± 0.10	8.23 ± 0.24	7.36 ± 0.15	6.84 ± 0.10	$7.92^* \pm 0.61$
PGFS	**7.81** ± 0.03	**7.16** ± 0.09	**8.96** ± 0.04	**7.63** ± 0.09	**7.15** ± 0.08	**8.93** ± 0.05



(a) (b) (c)

Figure 4. Performance comparison between Random Search (blue) and PGFS (orange) using CCR5 QSAR-based score as a reward (a): box plot of the QSAR-based CCR5 score per step of the iterative 5-step virtual synthesis. The first step (Reaction Step =0) in the box plot shows the scores of the fixed 2000 initial reactants (R1s). (b): distribution of the maximum QSAR-based rewards over 5-step iterations without the Applicability Domain (AD) filtering. (c): distributions of the maximum QSAR-based rewards over 5-step iterations after compounds that do not satisfy AD criteria of the corresponding QSAR model were filtered out from both sets.

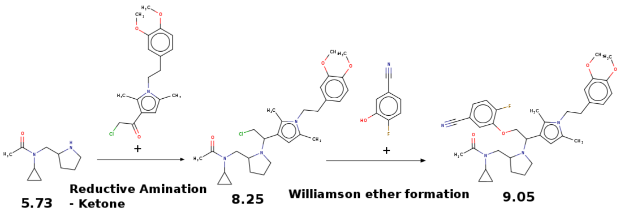


Figure 5. Structure of the compound generated by PGFS with the highest predicted activity against CCR5 and synthesis path used by the model. Predicted pIC50 values against CCR5 target are depicted under each structure at every reaction step.

5. Conclusion and Future Work

In this work, we introduce the first application of RL for forward synthesis in de novo drug design, PGFS, to navigate in the space of synthesizable small molecules. We use hierarchically organized actions where the second action is computed in a continuous space that is then transformed into the best valid reactant by the environment. PGFS achieves state-of-the art performance on QED and penalized clogP tasks. We also demonstrate the superiority of our approach in an in-silico scenario that mimics the drug discovery process. PGFS shows stable learning across all the tasks used in this study and shows significant enrichment in high scoring generated compounds when compared with existing benchmarks.

In future work, we propose to use a second policy gradient that solely updates the f network based on the value of its corresponding critic to efficiently learn to select transformation templates (unimolecular reactions) and to stop when the expected maximum reward in an episode is attained. Furthermore, one can use any RL algorithm for continuous action space like SAC (Haarnoja et al. (2018)) or a hybrid of a traditional planning and RL algorithm (V. et al. (2018); Anthony et al. (2017)). These future developments could potentially enable a better exploration of the chemical space.

Acknowledgements

The authors would like to thank Mohammad Amini for thoroughly reviewing the manuscript and Harry Zhao, Sitao Luan and Scott Fujimoto for useful discussions and feedback. 99andBeyond would like to thank Youssef Bennani for his valuable support and Davide Sabbadin for medicinal chemistry advice. SC is supported by a Canada CIFAR AI Chair and an NSERC Discovery Grant.

References

Anthony, T., Tian, Z., and Barber, D. Thinking fast and slow with deep learning and tree search. *CoRR*, abs/1705.08439, 2017. URL http://arxiv.org/abs/1705.08439.

Arts, E. J. and Hazuda, D. J. Hiv-1 antiretroviral drug therapy. *Cold Spring Harbor perspectives in medicine*, 2 (4):a007161, 2012.

Assouel, R., Ahmed, M., Segler, M. H., Saffari, A., and Bengio, Y. Defactor: Differentiable edge factorization-based probabilistic graph generation. *arXiv preprint arXiv:1811.09766*, 2018.

Balaban, A. T. Highly discriminating distance-based topological index. *Chemical Physics Letters*, 89(5):399–404, 1982.

Barth-Maron, G., Hoffman, M. W., Budden, D., Dabney, W., Horgan, D., TB, D., Muldal, A., Heess, N., and Lillicrap, T. Distributional policy gradients. In *International Conference on Learning Representations*, 2018.

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

Bradshaw, J., Paige, B., Kusner, M. J., Segler, M. H. S., and Hernández-Lobato, J. M. A model to search for synthesizable molecules. *CoRR*, abs/1906.05221, 2019.

Brown, N., McKay, B., Gilardoni, F., and Gasteiger, J. A graph-based genetic algorithm and its application to the multiobjective evolution of median molecules. *Journal of chemical information and computer sciences*, 44(3): 1079–1087, 2004.

Brown, N., Fiscato, M., Segler, M. H., and Vaucher, A. C. Guacamol: benchmarking models for de novo molecular design. *Journal of chemical information and modeling*, 59(3):1096–1108, 2019.

Butler, K. T., Davies, D. W., Cartwright, H., Isayev, O., and Walsh, A. Machine learning for molecular and materials science. *Nature*, 559(7715):547–555, 2018. doi: 10.1038/s41586-018-0337-2. URL https://doi.org/10.1038/s41586-018-0337-2.

Button, A., Merk, D., Hiss, J. A., and Schneider, G. Automated de novo molecular design by hybrid machine intelligence and rule-driven chemical synthesis. *Nature machine intelligence*, 1(7):307–315, 2019.

Cherkasov, A., Muratov, E. N., Fourches, D., Varnek, A., Baskin, I. I., Cronin, M., Dearden, J., Gramatica, P., Martin, Y. C., Todeschini, R., et al. Qsar modeling: where have you been? where are you going to? *Journal of medicinal chemistry*, 57(12):4977–5010, 2014.

Coley, C. W., Green, W. H., and Jensen, K. F. Machine Learning in Computer-Aided Synthesis Planning. *Accounts of Chemical Research*, 51(5):1281–1289, May 2018a. ISSN 0001-4842. doi: 10.1021/acs.accounts.8b00087.

Coley, C. W., Rogers, L., Green, W. H., and Jensen, K. F. Scscore: synthetic complexity learned from a reaction corpus. *Journal of chemical information and modeling*, 58(2):252–261, 2018b.

Coley, C. W., Eyke, N. S., and Jensen, K. F. Autonomous discovery in the chemical sciences part ii: Outlook. *Angewandte Chemie International Edition*, 2019a.

Coley, C. W., Thomas, D. A., Lummiss, J. A. M., Jaworski, J. N., Breen, C. P., Schultz, V., Hart, T., Fishman, J. S., Rogers, L., Gao, H., Hicklin, R. W., Plehiers, P. P., Byington, J., Piotti, J. S., Green, W. H., Hart, A. J., Jamison, T. F., and Jensen, K. F. A robotic platform for flow synthesis of organic compounds informed by AI planning. *Science*, 365(6453):eaax1566, August 2019b. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.aax1566.

Dulac-Arnold, G., Evans, R., Sunehag, P., and Coppin, B. Reinforcement learning in large discrete action spaces. *ArXiv*, abs/1512.07679, 2015.

Elton, D. C., Boukouvalas, Z., Fuge, M. D., and Chung, P. W. Deep learning for molecular design—a review of the state of the art. *Mol. Syst. Des. Eng.*, 4:828–849, 2019. doi: 10.1039/C9ME00039A.

Ertl, P. and Schuffenhauer, A. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of cheminformatics*, 1(1):8, 2009.

Fujimoto, S., van Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. *CoRR*, abs/1802.09477, 2018.

Gao, W. and Coley, C. W. The synthesizability of molecules proposed by generative models. *Journal of Chemical Information and Modeling*, Apr 2020. ISSN 1549-960X. doi: 10.1021/acs.jcim.0c00174. URL http://dx.doi.org/10.1021/acs.jcim.0c00174.

Gaulton, A., Hersey, A., Nowotka, M., Bento, A. P., Chambers, J., Mendez, D., Mutowo, P., Atkinson, F., Bellis, L. J., Cibrián-Uhalte, E., et al. The chembl database in 2017. *Nucleic acids research*, 45(D1):D945–D954, 2017.

Goh, G. B., Hodas, N. O., and Vishnu, A. Deep learning for computational chemistry. *Journal of Computational Chemistry*, 38(16):1291–1307, 2017. doi: 10.1002/jcc.24764.

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., and Aspuru-Guzik, A. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

Guimaraes, G. L., Sanchez-Lengeling, B., Outeiral, C., Farias, P. L. C., and Aspuru-Guzik, A. Objective-Reinforced Generative Adversarial Networks (ORGAN) for Sequence Generation Models. May 2017. URL https://arxiv.org/abs/1705.10843.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *CoRR*, abs/1801.01290, 2018.

Hartenfeller, M., Zettl, H., Walter, M., Rupp, M., Reisen, F., Proschak, E., Weggen, S., Stark, H., and Schneider, G. Dogs: reaction-driven de novo design of bioactive compounds. *PLoS computational biology*, 8(2):e1002380, 2012.

James, C., Weininger, D., and Delany, J. Smarts theory. daylight theory manual, 2000.

Jensen, J. H. 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.

Jin, W., Barzilay, R., and Jaakkola, T. Junction tree variational autoencoder for molecular graph generation. 2018.

Kier, L. B., Hall, L. H., et al. *Molecular structure description*. Academic, 1999.

Konze, K. D., Bos, P. H., Dahlgren, M. K., Leswing, K., Tubert-Brohman, I., Bortolato, A., Robbason, B., Abel, R., and Bhat, S. Reaction-based enumeration, active learning, and free energy calculations to rapidly explore synthetically tractable chemical space and optimize potency of cyclin-dependent kinase 2 inhibitors. *Journal of chemical information and modeling*, 59(9):3782–3793, 2019.

Korovina, K., Xu, S., Kandasamy, K., Neiswanger, W., Poczos, B., Schneider, J., and Xing, E. P. ChemBO: Bayesian Optimization of Small Organic Molecules with Synthesizable Recommendations. *arXiv:1908.01425 [physics, stat]*, August 2019. URL http://arxiv.org/abs/1908.01425. arXiv: 1908.01425.

Krenn, M., Häse, F., Nigam, A., Friederich, P., and Aspuru-Guzik, A. Selfies: a robust representation of semantically constrained graphs with an example application in chemistry. 2019.

Landrum, G. Rdkit: Open-source cheminformatics software. 2016. URL https://github.com/rdkit/rdkit/releases/tag/Release_2016_09_4.

Law, J., Zsoldos, Z., Simon, A., Reid, D., Liu, Y., Khew, S. Y., Johnson, A. P., Major, S., Wade, R. A., and Ando, H. Y. Route Designer: A Retrosynthetic Analysis Tool Utilizing Automated Retrosynthetic Rule Generation. *J. Chem. Inf. Model.*, 49(3):593–602, March 2009. ISSN 1549-9596. doi: 10.1021/ci800228y. URL http://dx.doi.org/10.1021/ci800228y.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N. M. O., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *CoRR*, abs/1509.02971, 2015.

Liu, S., Demirel, M. F., and Liang, Y. N-gram graph: Simple unsupervised representation for graphs, with applications to molecules. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8464–8476. Curran Associates, Inc., 2019.

Muratov, E. N., Bajorath, J., Sheridan, R. P., Tetko, I. V., Filimonov, D., Poroikov, V., Oprea, T. I., Baskin, I. I., Varnek, A., Roitberg, A., et al. Qsar without borders. *Chemical Society Reviews*, 2020.

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

Olivecrona, M., Blaschke, T., Engkvist, O., and Chen, H. Molecular de-novo design through deep reinforcement learning. *Journal of cheminformatics*, 9(1):48, 2017.

Patel, H., Bodkin, M. J., Chen, B., and Gillet, V. J. Knowledge-based approach to de novo design using reaction vectors. *Journal of chemical information and modeling*, 49(5):1163–1184, 2009.

Polykovskiy, D., Zhebrak, A., Sanchez-Lengeling, B., Golovanov, S., Tatanov, O., Belyaev, S., Kurbanov, R., Artamonov, A., Aladinskiy, V., Veselov, M., et al. Molecular sets (moses): a benchmarking platform for molecular generation models. *arXiv preprint arXiv:1811.12823*, 2018.

Popova, M., Isayev, O., and Tropsha, A. Deep reinforcement learning for de novo drug design. *Science advances*, 4(7): eaap7885, 2018.

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

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

Sanchez-Lengeling, B. and Aspuru-Guzik, A. Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, 361(6400):360–365, July 2018. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.aat2663. URL http://science.sciencemag.org/content/361/6400/360.

Schneider, G. Automating drug discovery. *Nature Reviews Drug Discovery*, 17(2):97–113, February 2018. ISSN 1474-1784. doi: 10.1038/nrd.2017.232. URL https://www.nature.com/articles/nrd.2017.232.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv*, 2017.

Segler, M. H., Kogej, T., Tyrchan, C., and Waller, M. P. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS central science*, 4 (1):120–131, 2017.

Segler, M. H. S., Preuss, M., and Waller, M. P. Planning chemical syntheses with deep neural networks and symbolic ai. *Nature*, 555(7698):604–610, 2018. doi: 10.1038/nature25978.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. Deterministic policy gradient algorithms. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pp. I–387–I–395. JMLR.org, 2014.

Simonovsky, M. and Komodakis, N. Graphvae: Towards generation of small graphs using variational autoencoders. In *International Conference on Artificial Neural Networks*, pp. 412–422. Springer, 2018.

Szymkuc, S., Gajewska, E. P., Klucznik, T., Molga, K., Dittwald, P., Startek, M., Bajczyk, M., and Grzybowski, B. A. Computer-Assisted Synthetic Planning: The End of the Beginning. *Angew. Chem., Int. Ed.*, 55(20):5904–5937, 2016. ISSN 1521-3773. doi: 10.1002/anie.201506101.

Tropsha, A. Best practices for qsar model development, validation, and exploitation. *Molecular informatics*, 29 (6-7):476–488, 2010.

V., S. K. G., Goyette, K., Chamseddine, A., and Considine, B. Deep pepper: Expert iteration based chess agent in the reinforcement learning setting. *CoRR*, abs/1806.00683, 2018. URL http://arxiv.org/abs/1806.00683.

Vinkers, H. M., de Jonge, M. R., Daeyaert, F. F., Heeres, J., Koymans, L. M., van Lenthe, J. H., Lewi, P. J., Timmerman, H., Van Aken, K., and Janssen, P. A. Synopsis: synthesize and optimize system in silico. *Journal of medicinal chemistry*, 46(13):2765–2773, 2003.

Walters, W. P. Virtual Chemical Libraries. *Journal of Medicinal Chemistry*, August 2018. ISSN 0022-2623. doi: 10.1021/acs.jmedchem.8b01048. URL https://doi.org/10.1021/acs.jmedchem.8b01048.

Walters, W. P. and Murcko, M. Assessing the impact of generative ai on medicinal chemistry. *Nature Biotechnology*, pp. 1–3, 2020.

Winter, R., Montanari, F., Steffen, A., Briem, H., Noé, F., and Clevert, D.-A. Efficient multi-objective molecular optimization in a continuous latent space. *Chemical science*, 10(34):8016–8024, 2019.

Wu, Y., Mansimov, E., Liao, S., Grosse, R. B., and Ba, J. Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. *NIPS*, 2017.

You, J., Liu, B., Ying, R., Pande, V. S., and Leskovec, J. Graph convolutional policy network for goal-directed molecular graph generation. *CoRR*, abs/1806.02473, 2018a. URL http://arxiv.org/abs/1806.02473.

You, J., Liu, B., Ying, Z., Pande, V., and Leskovec, J. Graph convolutional policy network for goal-directed molecular graph generation. In *Advances in neural information processing systems*, pp. 6410–6421, 2018b.

Zhou, Z., Kearnes, S. M., Li, L., Zare, R. N., and Riley, P. Optimization of molecules via deep reinforcement learning. *CoRR*, abs/1810.08678, 2018. URL http://arxiv.org/abs/1810.08678.