LibINVENT: Reaction-based Generative Scaffold Decoration for in Silico Library Design

(article starts on next page)

LibINVENT: Reaction-based Generative Scaffold Decoration for *in Silico* Library Design

Vendy Fialková, Jiaxi Zhao, Kostas Papadopoulos, Ola Engkvist, Esben Jannik Bjerrum, Thierry Kogej, and Atanas Patronov*

Cite This: https://doi.org/10.1021/acs.jcim.1c00469

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: Because of the strong relationship between the desired molecular activity and its structural core, the screening of focused, core-sharing chemical libraries is a key step in lead optimization. Despite the plethora of current research focused on *in silico* methods for molecule generation, to our knowledge, no tool capable of designing such libraries has been proposed. In this work, we present a novel tool for *de novo* drug design called LibINVENT. It is capable of rapidly proposing chemical libraries of compounds sharing the same core while maximizing a range of desirable properties. To further help the process of designing focused libraries, the user can list specific chemical reactions that can be used for the library creation. LibINVENT is therefore a flexible tool for generating virtual chemical libraries for lead optimization in a broad range of scenarios. Additionally, the shared core ensures that the compounds in the library are similar, possess desirable properties, and can also be synthesized under the same or similar conditions. The LibINVENT code is freely available in our public repository at https://github.com/MolecularAI/Lib-INVENT. The code necessary for data preprocessing is further available at: https://github.com/MolecularAI/Lib-INVENT-dataset.
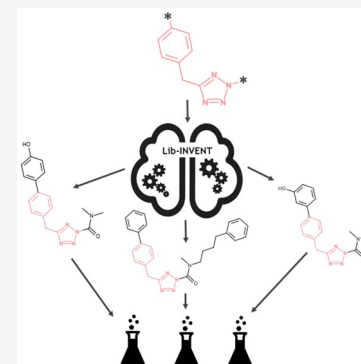
INTRODUCTION

With the recent advances in deep-learning techniques, such techniques are becoming increasingly popular tools in a range of areas—from automated vehicles to medicinal chemistry.[1,2] This is especially true for drug discovery, where the symbiosis between machine-learning models and human experts has the potential to significantly speed up the process of early drug discovery.[3] Because of their generalization abilities, deep generative models have become the core engine in most recent *de novo* design tools.[4,5] Despite the progress in the field of deep learning, such tools are still in the early stages of development, as they are adapting to satisfy the more specific needs of drug design.[6]

One of these specific requirements is in the lead optimization stage, when aiming to use focused libraries of small molecules to identify a promising lead compound.[7,8] In general, the purpose of lead optimization is to retain the favorable properties of the compound while optimizing the properties that still prevent the compound from becoming a drug candidate.[9] Because the desired activity is normally tied up to a given scaffold,[10] this use case boils down to retaining a certain molecular core and varying only specific moieties to satisfy the complex demands for the properties of the candidate molecule.[7] In practice, this can be addressed by screening very focused libraries that share the same core.[11] As an ideal scenario when synthesizing such a library, it should be possible to introduce the proposed moieties *via* the same or similar reactions to ensure that the reactions can be carried out under the same conditions. Related investigations

have been conducted on a much smaller scale in the works on matched molecular pairs[12] and fragment linking;[7] however, the explorations have not been previously extended to library generation or to considered chemical reactions.

For the purpose of this Article, we define a chemical library as follows:

Definition 1: Given a scaffold, *s*, the library is a set of molecules with the following conditions:

1. All include substructures
2. All molecules are accessible by the same sequence of synthetically relevant chemical transformations

In this Article, we propose a solution based on a *de novo* generative model capable of addressing the use cases outlined above. The model is built on the REINVENT[13] framework where a pretrained prior with a knowledge of general chemical syntax is focused *via* reinforcement learning to generate compounds optimized for a specific, user-defined task. In this work, we extend the objective from a single compound design to a library design. Specifically, the model can suggest moieties to decorate an input scaffold with a variable number of attachment
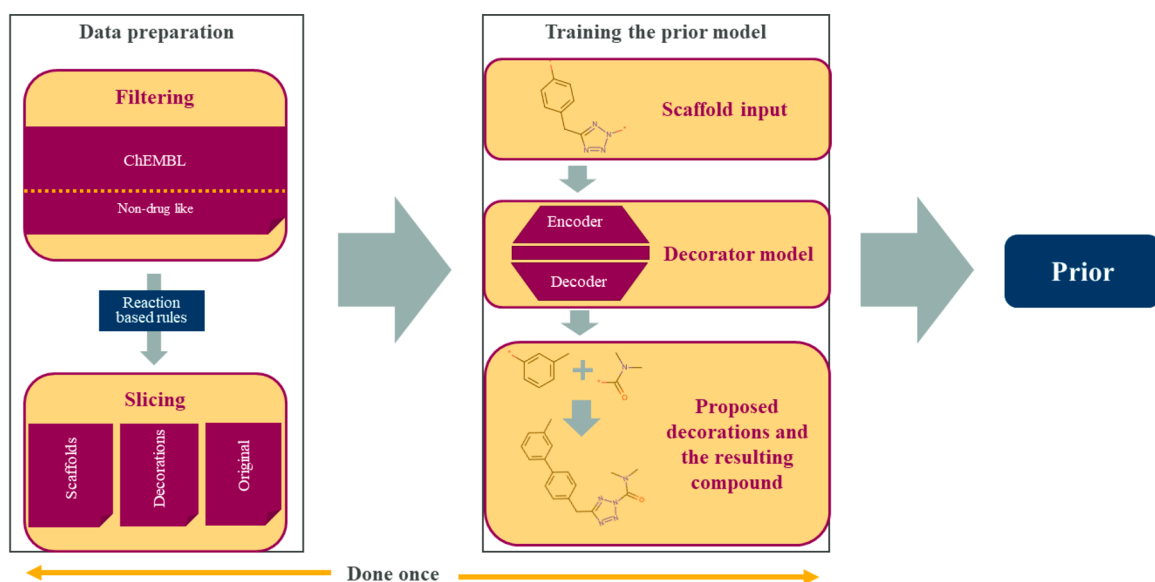
a: Dataset preparation and prior training. This is only performed once.
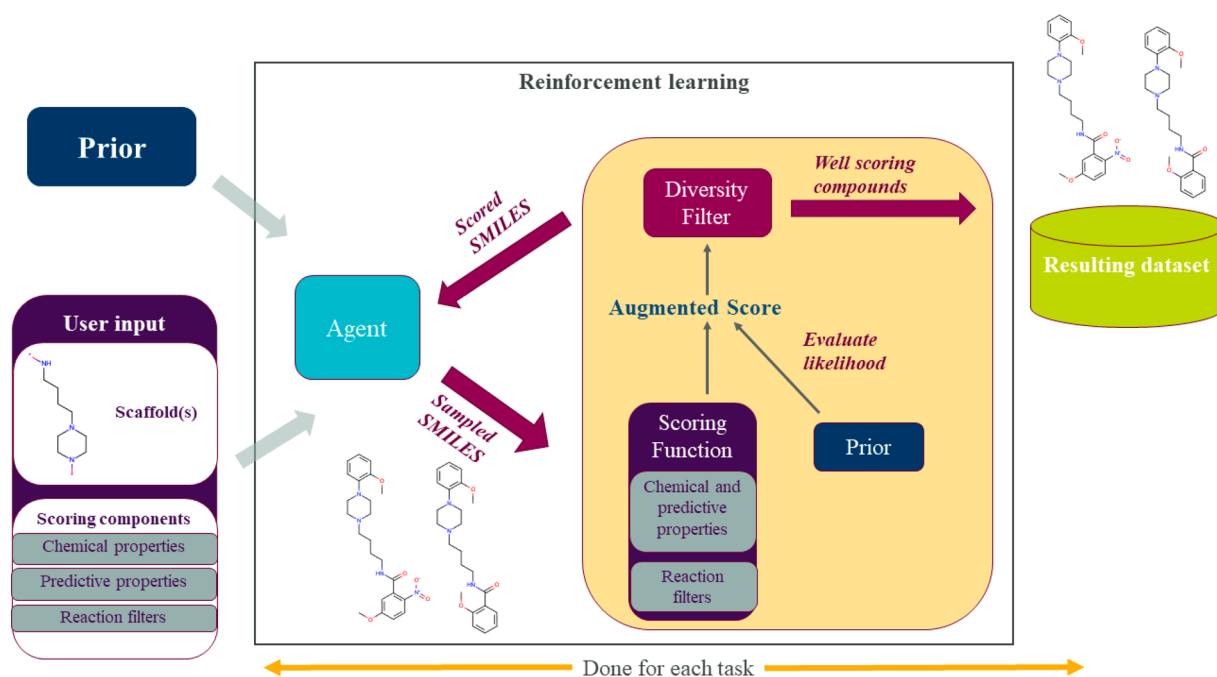
b: Production of a focused library in a reinforcement learning loop.

Figure 1. General workflow of the model. In the first stage (a), a generative decorator is trained on the preprocessed data. This model, called *prior*, is subsequently used in a reinforcement learning loop to generate a library for any user-defined task, as illustrated in panel b. It serves as both an initialization of the agent and a regularization component in the scoring function.

points for these decorations. In addition, the model can be put in a reinforcement learning (RL) scenario to learn to maximize a user-defined set of objectives. The resulting ideas will therefore be focused according to specific lead optimization goals. In contrast with prior work on scaffold decoration, these goals may include a set of reactions assigned to each attachment point of the scaffold so that the model learns to produce moieties attachable to that specific attachment point, in agreement with the given reactions. This way of generating chemical libraries gives the user a significant level of control over the output, enabling them to focus the model's creativity and leverage prior knowledge.[14] Satisfying condition 2 of the library definition further means that the generated library is more suitable for automation in the design and execution stage by reducing the number of reagents and reactions required in synthesis, provided that the specified reactions are selected to be amenable to automated synthesis. This further allows the chemist to optimally select reactions with a desired profile, which includes but is not limited to considerations of efficiency, literature coverage, or safety. Thus the number of DMTA (design−make−test−analyze) cycles required in the drug discovery process

decreases, improving the productivity of the incorporation of a generative model in the lead optimization pipeline.

The original REINVENT algorithm[13] proposes compounds optimized for solving a specific user-defined objective, and the recent GraphINVENT extends this to work to molecular graphs.[15] The algorithm introduced here, called LibINVENT, takes the work further and closer toward the utilization of chemistry automation platforms by building focused, easy synthesizable libraries. Related models have appeared in the literature over the recent years, focusing on the scaffold decoration itself or the usage of RL to guide the decorative process.[14,16,17] The major enhancement that LibINVENT brings to these methods lies in the volume and diversity of its output within a focused chemical space. Crucially, the fact that the generated libraries can be produced from the same starting scaffold using specific chemical reactions facilitates the uptake of the ideas in a wet lab environment and contributes to the possibility of automation of the drug-design process. By focusing on the design and synthesis of libraries instead of single molecules, the learning in each DMTA cycle can be increased, and accordingly, there is a need for fewer cycles to reach a clinical candidate.

■ GENERAL WORKFLOW

This section gives a high-level overview of the individual steps of the data preparation and model training and the usage of the algorithm to optimize various user-defined objectives. Figure 1 shows an overview of the workflow. Specific technical details will be further discussed in the Methods section. The motivation for the choices made in both the data preparation and the model design is further explained in the Discussion.

The model is a recurrent neural network (RNN) that takes a scaffold as an input and returns complete compounds obtained by attaching decorations to the input scaffold. There are two stages in the training: First, a general model is trained to learn the syntax of the SMILES language. We will henceforth refer to this model as the prior and stress that the training of the prior is not specific to a particular task and thus only occurs once. The second step corresponds to the general usage of this model, which is analogous to REINVENT: The prior is focused to solve a user-defined objective. In the case of LibINVENT, this is achieved through RL and might involve a requirement for specific chemical reaction types. Starting from a scaffold of interest, the general prior thus rapidly adapts to propose a focused chemical library consisting of thousands of compounds sharing a scaffold and chemical properties. Importantly, the generated compounds are collected during the RL process and not after, meaning that the model is typically no longer used after the completion of the RL run.

Data Preparation. Compounds from the publicly available ChEMBL Database, version 27,[18] represented by SMILES strings, were used to train the prior model. This choice of representing the chemical compounds by sequences of characters has several advantages and is common in the cheminformatics literature.[20] First, despite losing a certain level of chemical information,[11] this representation is significantly more memory efficient than the use of molecular graph data while implicitly retaining the molecular graph structure. Moreover, the SMILES strings are compatible with the chemical reactions expressed using the SMARTS language. This is crucial in the context of this work, which focuses on directly incorporating knowledge of chemical pathways into the generative model.

As is standard for computational applications in drug discovery, the first step of the data preparation process involves data purging and sanitization.[19] The purpose of purging is to remove undesirable compounds and outliers from the data set.[6] These, among others, include molecules containing rare SMILES tokens or elements that the model is unlikely to be able to learn and thus merely pollute the model's vocabulary, molecules with extremely large or low molecular weights, and salts, which are neither drug-like nor chemically friendly. Approximately 25% of the compounds present in the database are removed in this stage. For details of the implementation and filter criteria and the exact number of molecules present in the data sets, see the Supporting Information.

The second preprocessing step necessary to train a scaffold decorating model is compound slicing. There are many ways of slicing a molecule to obtain scaffold−decoration pairs for training a decorator model.[20] Recently, the exhaustive slicing of single bonds according to RECAP[21] rules has been explored.[16,17] Whereas this approach appears natural at a glance, it is not always effective for a wet lab chemist attempting to synthesize the proposed compounds.[22] Whereas the default RECAP rules, which include 11 bond cleavage types (amide, amine, ester, urea, ether, olefin, quaternary nitrogen, aliphatic carbon with aromatic nitrogen, lactam nitrogen with aliphatic carbon, aromatic carbon with aromatic carbon, and sulphonamide), aim to slice the molecules and identify preferred structural motifs, they still leave some cleaving possibilities out. The ability to follow real chemical reactions when decorating the scaffold is crucial; our experiments demonstrate that training on data sliced according to RECAP rules does not teach the prior to understand these chemical principles. This means that the model is unable to satisfy reaction requirements when designing chemical libraries.

Practical synthesis and chemical considerations should thus be taken into account when slicing the molecules to ensure that the reverse process (forward synthesis) is synthetically valid.[23] In their recent paper, Horwood and Noutahi[24] propose incorporating chemical synthesis routes directly into the model by designing a *de novo* generator based on chemical reactions. Given the starting reactants, their model proposes drug-like molecules by selecting other appropriate reactants and specific reactions used to transform and connect the molecules into a resulting compound. This novel approach significantly improves the synthesizability of the proposed molecules; however, it still lacks the ability to design libraries and the degree of flexibility and generality that is desirable in *de novo* generators. Specifically, training has to occur on a data set relevant to the final task at hand, and there is a limited capacity for knowledge transfer and extension to more specific tasks without retraining.

In this work, an alternative data preprocessing approach to the one used by Arús-Pous *et al.*[25] is proposed to build a knowledge of chemical reactions directly on the training data set composed of the filtered ChEMBL database. Thirty-seven handcrafted reaction-based rules are used to slice the training compounds into scaffolds and decorations so that each split is a result of a known, easily implementable chemical reaction. A complete list of the reaction SMIRKS can be found in the tutorial section of our LibINVENT data set public repository; moreover, the Supporting Information provides details of the exact steps taken and the number of compounds, scaffolds, and decorations used at each step. We demonstrate that the reaction-based slicing method enables the generative model to propose decorations according to the chemical reactions used in training. The output
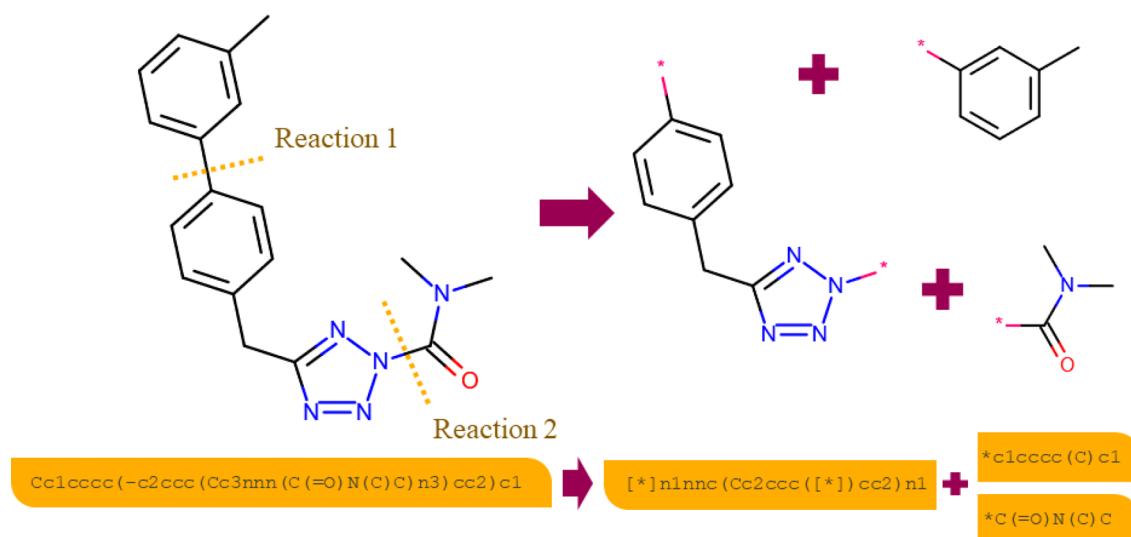
Figure 2. Example of a sliced molecule resulting in a scaffold with two attachment points and the corresponding decorations.

therefore benefits from high validity and a better likelihood of being synthetically feasible. An illustration of the process is provided in Figure 2.

Model Training. The prior model is trained using the teacher forcing algorithm[26] to maximize the conditional likelihoods of the generated compounds given the scaffolds. Even a single pass through the data set teaches the model to generate chemically valid SMILES strings; the optimal state balancing the coverage of the chemical space and the overfitting is, however, reached after approximately eight epochs. At each epoch, a different randomized representation of the training and validations SMILES is used, as this further prevents overfitting.[27] As previously mentioned, it is crucial to note that this training needs to happen only once because the prior can be reused for a wide range of tasks without the additional transfer learning stages that were often required in the previously introduced models.[28] The resulting prior model is provided in our public GitHub repository along with the code and data necessary to reproduce the training.

Case-Specific Usage: Focusing a Prior *via* Reinforcement Learning. The case-specific usage of the model involves focusing the prior on a specific task. This fine-tuning is efficiently achieved by setting up an RL loop in which the prior iteratively proposes compounds and receives task-specific rewards for its output. During the run, all high-scoring compounds are stored in a virtual chemical library that we will refer to as the *resulting data set*; the production of the library therefore begins instantaneously once an RL run is set up and continues throughout the training of the RL agent.

The rewards are shaped by a scoring function defining desirable chemical or structural properties to guide the model toward producing compounds of interest.[8] However, because the objective is to explore a rather narrow space of solutions (molecules) designed for a given scaffold, this may lead to a mode collapse.[29] To achieve a stable RL process, we introduce a mechanism that relies on diversity filters (DFs) previously described by Blaschke *et al.*[13] DFs and the prior likelihoods of the proposed compounds can be included when calculating the reward. DFs penalize the RL agent for repeatedly generating the same compound, which significantly reduces the risk of mode collapse toward a single high-scoring solution (molecule). The prior likelihood serves as an additional regularizer, anchoring the

agent to the previously learnt chemical space and ensuring that the SMILES syntax is not forgotten.[30]

Another reward-modifying factor is the reaction filter (RF). The introduction of RFs to the learning process means that the proposed libraries can be synthesized using selected reactions, facilitating the creation of focused libraries. RFs are designed to be selective, so that a reaction or a set of reactions can be specified for each attachment point of the scaffold. This gives the user significant control over the output of the model and enables leveraging prior chemical knowledge. The full practical implementation of the RL procedure is described along with its mathematical background in the section Focusing the Prior via Reinforcement Learning. A number of reaction definitions is further published in our public repository.

We emphasize that focusing the pretrained prior using RL makes the LibINVENT decorator model widely applicable to a variety of real-world scenarios with a range of reactants. Libraries containing thousands of high-scoring, synthesizable molecules can be obtained within minutes or tens of minutes, and the more expensive training of the prior model does not need to be repeated for new libraries.

■ METHODS

Model Architecture. The architecture of the model is analogous to the scaffold decorator introduced by Arús-Pous *et al.*[16] The decorator model uses an encoder−decoder architecture where both the encoder and decoder are RNNs with three hidden layers of dimension 512, and the embedding is of size 256. During training, dropout at rate 0.1 has been used.

We refer to the collection of tokens recognized and used by the model as the vocabulary. This is composed of all of the SMILES characters present in the pruned training data set and enriched by the special "END" and "START" tokens determining the beginning and ending of a SMILES string. The length of the vocabulary corresponds to the dimension of the multinomial distribution over which the tokens are sampled. For details of the tokens included, see the Supporting Information.

Validation Set. As in any machine-learning model, a good validation set is required to fairly evaluate the performance.[31] The objective of the prior model training is to learn to decorate scaffolds so that the resulting compounds lie in the drug-like

chemical space spanned by the training data set. This nature of the modeling objective affects the choices made when preparing the validation set. Whereas it is common to randomly hold out a portion of the training data and use these for evaluation,[32] the sliced data set used here does not lend itself well to this approach. To be able to fairly judge the generalization ability of the model in previously unseen scaffolds, it is necessary to ensure that the validation scaffolds are not present in the training data set. Optimally, even compounds structurally similar to these need to be removed from the training set to fairly validate the performance of the model.[33] At the same time, the general distribution of the validation scaffold properties should mimic that of the training set to evaluate how well the model learns to follow the data distribution.

With these considerations in mind, we handcrafted the training−validation split by selecting one scaffold with each number of attachment points at random. Because the compounds in the data set were sliced up to four times, the maximum possible number of attachment points is four. Then, all scaffolds sharing a Murcko scaffold[34] with one of the four compounds used to define the seed for the validation set are removed from the data set and used for validation. The choice to consider only the molecular cores consisting of ring structures stripped of side chains is motivated by the fact that the removed sets of compounds resemble the concept of "chemical series", as used by medicinal chemists.[10] Removing entire chemical series based on a specific Murcko scaffold thus naturally reduces the bias in the model evaluation and objectively tests its generalization ability.

We have selected the dopamine receptor D2 (DRD2) as the biological target of interest. This is a commonly used target in molecular generative models studies by our group[30,35,36,25] and other groups[24,37,38] in this field and allows access to large publicly available structure−activity relationship (SAR) data sets.
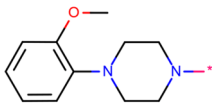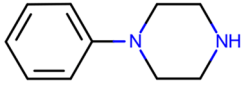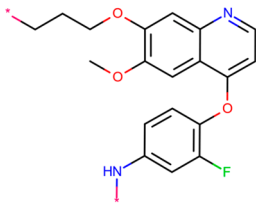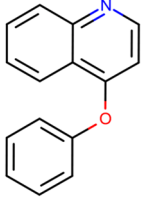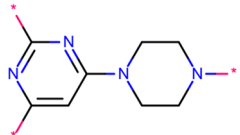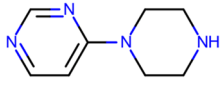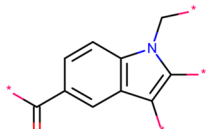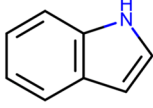
We remove all compounds sharing a scaffold with compounds found in the data set obtained by slicing the DRD2 scaffolds according to the set of reactions previously used to slice ChEMBL.[39] These compounds are not included in either the validation or the training set. In this way, the training and validation sets are kept independent, and the subsequent external validation on DRD2 remains unbiased.

Representative scaffolds for the held-out compounds used for validation are shown in Table 1 along with their Bemis−Murcko representations. The resulting validation set excluding the DRD2 data contained 241 137 unique entries. The training set contained the remaining 23 080 572 entries. These numbers show that the consideration of Bemis−Murcko scaffolds filters out a non-negligible number of compounds, very similar to the original held-out scaffold. At the same time, the size of this data set means that despite the validation representing only ∼1% of the data, sufficient information is still included to assess the generative ability of the decorator.

Until now, all SMILES have been canonicalized to ensure uniqueness. However, using different SMILES representations during the training of deep-learning models leads to improvements in the generalizability in activity modeling,[40] representation learning,[41] and SMILES generation. Before training the generative model, a different randomized representation of the training data set is obtained for each epoch of teacher forcing training. The same is applied to the validation set.

Pretraining the Prior *via* Teacher Forcing. As mentioned before, the training process of LibINVENT resembles the

Table 1. Held-Out Validation Scaffolds Picked from the Data and Their Bemis−Murcko Forms, Based on Which the Remainder of the Held-Out Scaffolds Were Chosen

Library scaffold	Bemis-Murcko scaffold
[*]N1CCN(c2ccccc2OC)CC1	
[*]CCCOc1cc2nccc(Oc3ccc(N[*])cc3F)c2cc1OC	
[*]N1CCN(c2cc([*])nc([*])n2)CC1	
[*]Cn1c([*])c([*])c2cc(C(=O)[*])ccc21	

training of REINVENT 2.0.[13] First, teacher forcing[42] is used to train the prior model capable of creating chemically valid compounds containing a given scaffold. In our case, the prior is an RNN taking a scaffold as an input and returning relevant decorations to connect to all available attachment points of the scaffold, much like the model recently introduced by Arús-Pous *et al.*[25] The number of resulting molecules corresponds to the batch size.

The generation process can be seen as a sequential conditional likelihood maximization problem. The output of the model represents a probability distribution over the token space containing all of the possible SMILES tokens present in the training data set enriched by the "START" and "END" tokens, given the scaffold and previously generated tokens in the decoration. The objective function to be maximized can thus be written as

$$J(\theta|S = s) = \prod_{\text{decoration points}} \left\{ P(X_1|S = s, \theta) \times \prod_{i=2}^{T} P(X_i = x_i|X_{i-1} = x_{i-1}, \ldots, X_1 = x_1, S = s, \theta) \right\}$$

(1)

Here θ represents the network parameters to be determined, X_i, where $i = 1, ..., T$ are the random variables corresponding to the tokens, whereas x_i are the observed (or in this case previously generated) tokens. Analogously, S and s refer, respectively, to the random variables corresponding to the input scaffold and the specific scaffold themselves. In this work, the scaffold is given *a priori*, and the distribution is therefore deterministic. Finally, T is another random variable that determines the length of the decoration SMILES string. In practice, we do not sample its distribution. Instead, the process ends when the "END" token is sampled.

The implementation and training details are described in the Supporting Information.

Focusing the Prior *via* Reinforcement Learning. *Motivation.* Because of the vastness of chemical space, it is typically not sufficient to be able to produce drug-like molecules; indeed, depending on the specific design objective, the exploration of a narrower chemical subspace is many times desirable, in particular, in lead optimization.[43] Specific focusing is thus a crucial step in developing a generative model capable of proposing compounds that are useful in a context like lead optimization. To achieve this, the parameters of the pretrained prior network need to be modified to target a narrower chemical subspace. At the same time, it has been observed that deviating too far from the prior can have catastrophic consequences where the model loses its knowledge of the valid SMILES syntax.[43,44]

To focus the model, an RL agent is initialized as a network with weights and architecture identical to those of the pretrained prior. To define the task, a reward function is constructed to guide the agent's learning, taking SMILES strings as input and returning scores in the range of $[0, 1]$. The function rewards compounds with desirable properties, promotes varied output through DFs, and specifies desired reactions to be used *via* RFs. Then, standard policy iteration RL is applied: In successive iterations, the agent proposes decorations for the scaffold and updates its parameters in a gradient ascent fashion based on the rewards these decorations receive. During the training, all syntactically valid compounds (SMILES strings) with a score exceeding a user-defined threshold are stored in the *resulting data set* and made available to the user at the end of the run. A successful run results in a constantly increasing data set because the model produces new relevant outputs at each step during the run. In an optimal scenario, the *resulting data set* increases linearly with the number of steps, with the gradient corresponding to the batch size. This motivates the following definition of a yield metric used to evaluate the degree of success of the runs

$$\text{yield} = \frac{|\text{resulting data set}|}{\text{batch size} \times \text{number of steps}} \quad (2)$$

The consideration of yield as opposed to the raw number of molecules produced is important because the produced numbers ultimately depend on the selected batch size. A model trained with a batch size of 32 returns twice as many compounds at each epoch as one with a batch size of 16. The important question, however, is how many of the 32 compounds are relevant and unique.

Mathematical Background. The starting point for a mathematical description of the RL procedure is defining a state space S_t and the corresponding action space $A_t(s_t)$ as well as rewards $r_t := R(a_t)$ for all $s_t \in S_t$, $a_t \in A_t$. In the context of molecule decoration, an action is a proposed decoration (or

decorations) for the scaffold, whereas the state contains information about all previously proposed decorations and the rewards assigned to those, that is, $s_t = \sum_{\tau=1}^{t-1} r_\tau$. Note the reward function R is fixed throughout the training.

At each step, the RL agent randomly samples an action (*i.e.*, proposes a decoration) according to its policy π_θ. The aim is to find the value of the parameters, θ, leading to an optimal policy, $\pi_\theta*$, maximizing the expected cumulative rewards across the whole run. In other words

$$\theta* = \text{argmax}_\theta \sum_{t=0}^{T} \mathbb{E}_{A \sim \pi_\theta}(R(A)|S_t = s_t) \quad (3)$$

The expected value is maximized at each time step in a greedy manner. The RL objective function at each step can therefore be written as

$$J(\theta) = \mathbb{E}_{A \sim \pi_\theta}(R(A)|S_t, \theta) \quad (4)$$

where the expectation is taken over the distribution of the actions.

Gradient ascent methods are typically used to maximize the objective. Exploiting the fact that $\nabla \log f(x) = \frac{\nabla f(x)}{f(x)}$, the gradient of eq 4 at step $t + 1$ can be written as

$$\nabla_\theta J(\theta) = \sum_{a \in A_{t+1}} R(a) \nabla_\theta \log \pi(A = a|S_t = s_t, \theta) \quad (5)$$

Equation 5 is the basis of many popular RL algorithms such as REINFORCE.[45] If the goal is to maximize the cumulative rewards across N training epochs, then it suffices to add an extra summation over all of the time steps, which results in a similar expression, the key feature of which is the fact that it is sufficient to compute the gradients of the log likelihoods of obtaining a gradient ascent update step.

Without further regularization or adjustments, these methods are known to suffer from high variance and instability.[46] In the case of molecular generation, however, the aim is to produce a large number of varied, interesting molecules.[24] This means that a certain level of variance is desirable to promote the exploration of the chemical space and to prevent mode collapse toward a single, high-scoring solution. Our experiments show that with an appropriate choice of the reward function, high variance does not hinder the models from producing relevant output.

Policy Iteration Rewards. A crucial requirement for a successful setup of an RL run is a good definition of the reward. In our case, this has to guide the agent in the right direction to solve the specific practical task and promote diversity. Similar to Blaschke *et al.*,[13] we investigate rewards assembled from a combination of two elements: the scoring function $S(a) \in [0,1]$ quantifying how well the proposed compound solves the task and the prior likelihood $\pi_{\theta_{\text{prior}}}(a) = \pi(a, \theta_{\text{prior}})$. Because the agent and prior share architecture, their likelihood functions differ only in the values of the parameters θ.

Scoring Function. The $S(a)$ itself is composed of multiple weighed elements that are summed or multiplied; the final score is then normalized to lie in the unit interval $[0,1]$. A range of components is supported from molecular descriptors such as the molecular weight, the topological polar surface area (TPSA), pretrained predictive models, docking,[47] and the ROCS similarity.[48] As previously mentioned, DFs and RFs may be imposed to further restrict the space of relevant output and to promote diversity.

The DF works by penalizing the model for producing an identical compound multiple times in a single batch. This is beneficial in preventing the agent from repeatedly proposing the same, high-scoring compound multiple times, which can lead to mode collapse.[49] A well-selected DF therefore balances the exploration and exploitation of the chemical space.

Finally, RFs are a feature that give the user greater control over the generated compounds. Two types of RFs are implemented: a general filter determining what reactions should occur to decorate the scaffold and a selective filter assigning the specific reactions to the individual attachment points. RFs use retrosynthetic reaction definitions that are compatible with RDKit.[50] Any valid reaction definition can be applied here; however, we provide, together with the code repository, a list of 37 predefined reactions that were used for the generation of training data. For each attachment point, the user can simultaneously provide a variable number of reactions within the same run. RFs serve as a penalty component; however, if any of the listed reactions is satisfied, then for the given attachment point, the score is not penalized. This requires a chemical understanding of the nature of the problem to avoid the situation where a nonfeasible reaction is required for a given attachment point; it is nevertheless a novel and efficient way of generating libraries of similar drug-like compounds that are readily synthesizable.

Different Reward Strategies. The motivation for the use of the prior likelihood in the reward function is identical to that of Olivecrona *et al.*[44] The pretraining ensures that the model is capable of generating valid SMILES of drug-like molecules. This serves as an anchor; it is desirable to discourage the agent from deviating too far from its prior state because a strong focus on maximizing the score can alone lead to either a mode collapse or a loss of generative ability altogether. Once the agent moves to a parameter space that does not lead to a valid SMILES syntax, it does not receive any rewards at all and cannot continue learning through gradient ascent.

On the basis of the above discussion, we follow previous work in defining the augmented log likelihood $\log\pi_A(a) = \log\pi_{\theta_{prior}}(a) + \sigma S(a)$. Here σ is a constant hyperparameter scaling the output in the same range. We note that the log likelihood is a monotonic increasing function that takes values in $(-\infty, 0)$, which means that the reward is a monotonic increasing function in $(-\infty, \sigma)$. In experimental setups, this likelihood is shown to serve well as a directional guide, leading the agent to more focused and interesting chemical spaces. The intuitive rationale for this is that the augmented likelihood balances the prior anchor with the task-specific objective.

Finally, four different RL learning strategies are proposed based on four different reward functions:

1. $R(a) = S(a)$. This method, henceforth referred to as **MASCOF** (**Ma**ximize **Sco**ring **F**unction), is a simple implementation of the standard REINFORCE algorithm where the scoring function directly serves as the reward.[45] This standard approach to solving an RL problem by maximizing the scoring function without anchoring it to the prior is a natural first step and can be seen as a baseline for the other methods. However, our experiments demonstrate that the RL agent struggles to remain in the valid chemical space without the anchor. Similar observations have been made in the past, typically arguing that the initial sparseness of rewards leads to the model struggling to begin learning.[49]

2. $R(a) = \log\pi_A(a)$. Because the augmented likelihood attempts to balance the prior likelihood and the scoring function, it can be seen as a reward itself. We call this method **MAULI** (**Ma**ximize **Au**gmented **Li**kelihood).

3. $R(a) = \log\pi_A(a) - \log\pi_\theta(a)$. This approach, dubbed **DAP** (**D**ifference between **A**ugmented and **P**osterior), can be shown to be equivalent to the strategy introduced by ref 44. Their work frames the RL slightly differently, focusing on loss minimization of the square loss between the augmented and posterior log likelihoods: $\mathcal{L}(\theta) = (\log\pi_A(a) - \log\pi_\theta(a))^2$. Whereas it is not a standard policy iteration approach, it does perform well in focusing the agent. For a full derivation of the equivalence of these two approaches, we refer the interested reader to the Supporting Information.

4. $R(a) = -(\log\pi_A(a) - \log\pi_\theta(a))^2$. Noting that the rationale behind the DAP strategy is minimization of the difference between the two likelihoods, the fact that the likelihoods are unbounded means that with the formulation in number 3, the reward may, in theory, keep increasing infinitely as the posterior probability approaches zero. In practice, this is rarely observed. For mathematical rigor, however, we define a final strategy called **SDAP** (**S**quared **D**ifference between **A**ugmented and **P**osterior). The negative of the squared loss is directly used as a reward function here, meaning that the agent is always encouraged to approach the augmented likelihood; maximizing the reward is equivalent to minimizing the square loss.

■ EXPERIMENTS

LibINVENT implements a novel deep-learning-based drug discovery approach for the generation of focused chemical libraries, given an input scaffold, by taking into account specific chemical reactions. This approach was designed to improve the productivity in the DMTA cycle through proposing a library of compounds that can be synthesized though the same chemical reactions. Thus more compounds can be synthesized with the same effort in an DMTA cycle and, accordingly, each DMTA cycle will be more informative.[51] We therefore introduce a range of experiments with the aim to demonstrate the potential of our proposed models to improve productivity. Specifically, we focus on promoting diversity of output, generating molecules that are readily synthesizable by a given reaction and determining R-group substitutions for lead optimization projects.

The objectives of the experiments are as follows:

- Determine the optimal learning strategy for the RL loop.
- Demonstrate the ability to follow specified reactions to decorate a given scaffold and contrast this with a model trained on a data set obtained using RECAP rules as opposed to reaction-based slicing.
- Demonstrate the ability to decorate scaffolds with various numbers of attachment points.

A baseline objective for the experiments is the generation of ligands that are optimized against DRD2. Two sets of tasks, based on two approaches to steer the model toward the desirable chemical space, have been executed. In the first set of experiments, a QSAR predictive model for the activity of the generated compounds is used as a component of the scoring function. This model is subsequently replaced by a 3D shape and pharmacophore similarity ROCS[52] scoring component to

promote the 3D similarity of the output to haloperidol, a known DRD2-active ligand. The details of the implementations can be found in the Supporting Information; our public repository further holds both the trained QSAR model and the ROCS input used. In all of the experiments, a DF is further added to the scoring function to promote variation in the output, along with custom alerts preventing the agent from proposing compounds with too-large rings and non-drug-like groups.

Figure 3 displays the testing scaffold. The choice is motivated by it being a good starting scaffold for generating DRD2 actives.
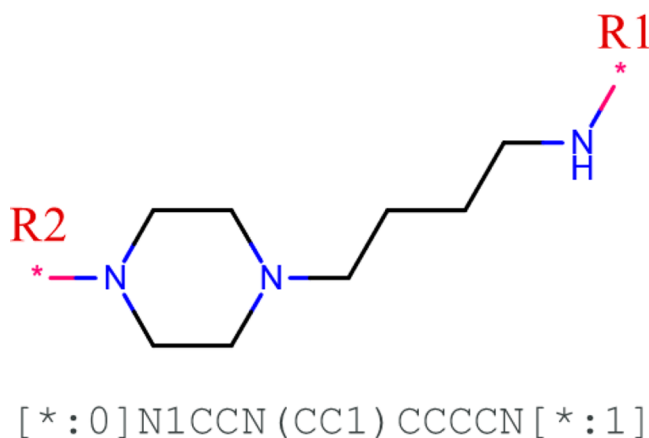

Figure 3. Testing scaffold. We note that in SMILES syntax, the decoration points are labeled by [*:0] and [*:1], which correspond to R2 and R1, respectively, in the molecular graph.

Furthermore, it has two attachment points, which is common in real-world applications. Whereas we demonstrate the ability of the LibINVENT decorator to work with scaffolds with up to four attachment points, library synthesis is most commonly executed on fewer attachments because this gives a better balance between the flexibility and the complexity of the library production step.

Evaluation Metrics. The complexity of the task of molecule generation means that the choice of a metric for model evaluation needs to be carefully considered to ensure that all relevant issues are addressed. For library generation, it is desirable to produce libraries based on user-defined criteria. Within these criteria, however, the libraries still differ in size, diversity, and scores achieved according to the scoring function. We have previously defined the yield metric, which helps evaluate what fraction of the generated ideas is scored above a given threshold. Nevertheless, this alone is not sufficient to give a fair comparison of the libraries.

We address the question of the diversity of the output *via* two approaches, as appropriate in the given scenario. To determine the effect of a change in the scoring function, we evaluate the overlap between the output libraries. This would show whether the methods produce significantly different results. When testing the effect of specific RFs, it may be more interesting to analyze the variation in the chemical properties of the proposed decorations for each attachment point. This smaller-scale view offers a more fine-grained picture of the level of control the user has over the design of their specific library.

Results. *Comparison of the Learning Strategies.* For each of the four learning strategies, two experiments are set up to contrast their abilities to propose molecules according to a given set of criteria. In the first experiment, only a QSAR predictive property is used. The motivation for this experiment is to

benchmark the abilities of the models to generate compounds when unconstrained by chemical reactions. The results of the experiments are displayed in Table 2, which gives an overview of the average results over three individual runs. For a detailed breakdown over the runs, refer to the Supporting Information.

Table 2. Comparison of the Four Learning Strategies for a QSAR Model with No Reaction Filters[a]

	number of compounds found	yield	average mean score in *resulting data set*
DAP	10510 ± 69	0.821 ± 0.005	0.722 ± 0.005
MAULI	8573 ± 271	0.670 ± 0.021	0.658 ± 0.015
MASCOF	4432 ± 50	0.346 ± 0.004	0.657 ± 0.022
SDAP	4755 ± 153	0.372 ± 0.012	0.695 ± 0.011

[a]Uncertainty boundaries correspond to the largest deviation from the mean observed over the three runs. We note that these are very low, showing a strong consistency between the trials. The yield metric is calculated as previously defined in eq 2.

In a second experiment, a selective RF is added to the scoring function. Attachment point R1 should be decorated using amide coupling, whereas the second attachment follows the Buchwald reaction. The exact implementation and SMIRKS definitions of the corresponding equations can be found in our public repository. The results of the experiment are shown in Table 3.

The numerical results show that in agreement with past observations,[44] the DAP learning strategy is the most successful strategy on multiple counts. First, the high average score of the compounds in the *resulting data set* for both runs indicates the ability of this model to consistently produce high-scoring molecules throughout the run. This is further supported by the size of the output and the correspondingly high yield: Even when selective RFs are applied, >80% of the proposed compounds have a score higher than the threshold of 0.4 chosen as the condition for inclusion in the *resulting data set*. Moreover, nearly 90% of the *resulting data set* compounds satisfy both of the RFs, which gives strong support for using this strategy in virtual chemical library creation.

Finally, to understand the training of the four respective strategies, we plot the average scores achieved at each step. It is crucial to note that thanks to the pretraining of the prior, a steeply increasing training curve is not expected to occur because the choice of the starting scaffold is task-specific and typically leads to high scores from the first iteration. The comparison is nevertheless a useful aid in the comparison because it further explains the process.

The evolution of the average scores across the runs is displayed in Figure 4. In both scenarios, the DAP strategy clearly outperforms the remaining optimization methods, quickly increasing from the starting point and then plateauing at the highest level. When RFs are introduced, this dominance becomes even more significant. As displayed in Figure 5, the DAP strategy is the only strategy capable of rapidly adapting to this requirement and satisfying these filters. The SDAP strategy also demonstrates learning but is much slower in adapting to the specific task. Both MASCOF and MAULI, on the contrary, decline slightly from the initial point, as they struggle to retain the prior knowledge of the chemical space, which is demonstrated by the dropping validity. No evidence of learning to follow the required reactions is apparent.

In both Figures 4 and 5, the shaded areas correspond to the minimum and maximum values achieved over the three runs,

Table 3. Results of the Four Learning Strategies When a QSAR Predictive Model and a Selective Reaction Filter Are Employed

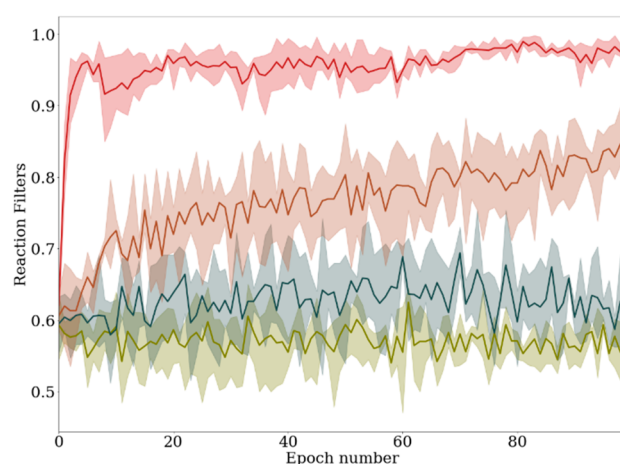|	number of compounds found	yield	average mean score in output	ratio of fully satisfied reaction filters
DAP	10454 ± 192	0.817 ± 0.015	0.729 ± 0.008	0.892 ± 0.032
MAULI	5179 ± 518	0.405 ± 0.012	0.564 ± 0.009	0.230 ± 0.027
MASCOF	2846 ± 854	0.222 ± 0.067	0.574 ± 0.030	0.297 ± 0.076
SDAP	4033 ± 302	0.315 ± 0.024	0.622 ± 0.019	0.457 ± 0.136
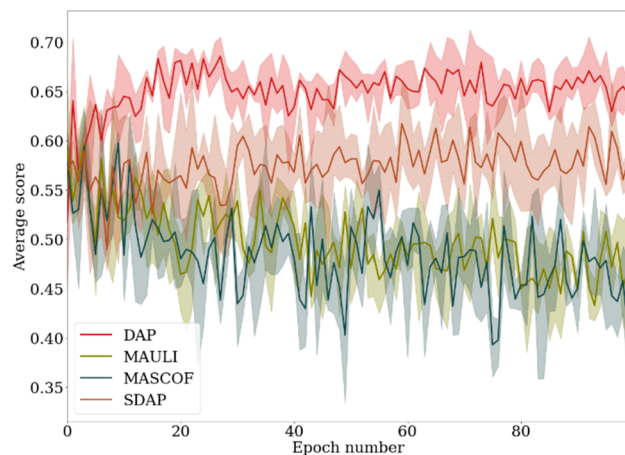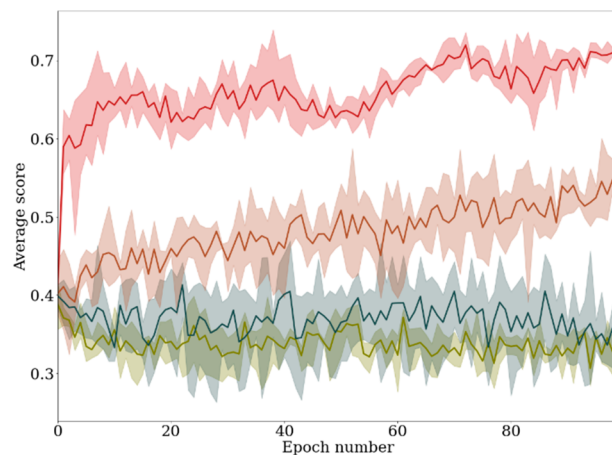
a: No reaction filters.   b: Amide coupling and Buchwald reaction filters.

Figure 4. Average score across a generated batch of compounds per epoch for each of the running strategies. The presence of reaction filters increases the superiority of the DAP strategy. Both MAULI and MASCOF are incapable of adapting the production to the user-defined objective.
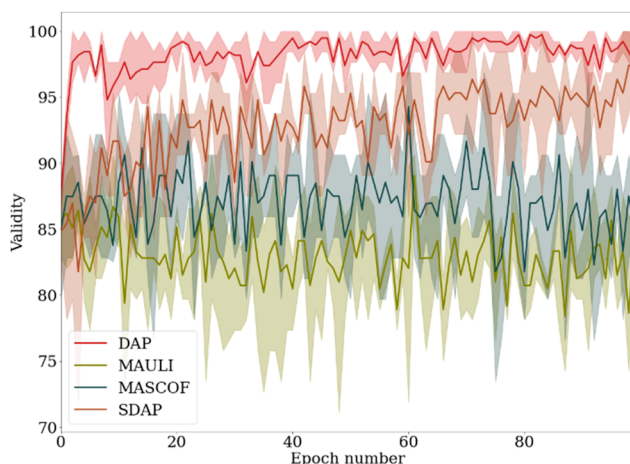
a: Validity of generated compounds.   b: Reaction Filter scores of generated compounds.

Figure 5. Validity of the output and reaction filter scores for each learning strategy when a reaction filter is imposed. The DAP learning strategy clearly outperforms the remaining approaches.

whereas the solid lines represent the mean. In addition to the expected stochasticity arising from the randomness in the optimization procedure, the plots indicate that the general behavior of the strategies is consistent across runs. This observation is in agreement with the previous analysis of numerical data. In the subsequent experiments, we therefore restrict all in-depth analysis to only a single run per model, as the stochasticity does not significantly affect the output, justifying the low levels of variance between runs by numerical tables. Moreover, because the analysis above shows a clear dominance of the DAP learning strategy, it is our method of choice in all of the subsequent experiments.

Comparison of Slicing Strategies. Reaction-based slicing used to preprocess the data set is one of the key aspects of this work. We therefore design a second set of experiments aimed at evaluating the effect of pretraining on data sliced according to chemical reactions, as opposed to RECAP rules when tackling RFs. To this end, we use the model of Arús-Pous *et al.* as an alternative against which to benchmark.[16] This model provides a fair comparison because its architecture and training procedure are exactly equivalent to those of our LibINVENT prior, with a crucial difference in data preparation.

Two experiments are conducted with these two priors. In both, the same RFs as before are imposed, decorating

attachment point 1 by amide coupling and attachment point 2 through the Buchwald reaction. The difference lies in the scoring function component, which uses a predictive QSAR model in the first experiment and replaces it by a ROCS 3D similarity scoring in the second task. The purpose of this change is to uncouple the effect of the scoring function from the RF and evaluate the effect of the preprocessing method as accurately as possible. The definition of the shape and pharmacophore ROCS query based on haloperidol is displayed in Figure 6.
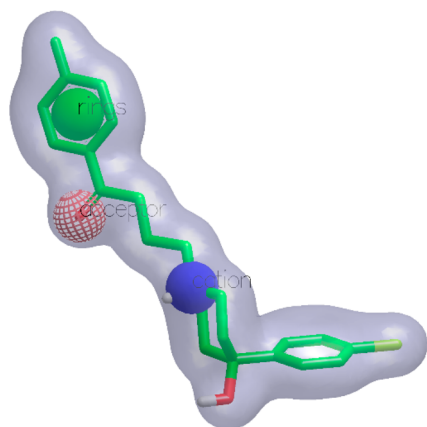

Figure 6. ROCS shape and pharmacophore query definition for haloperidol.

A numerical comparison of the experiments is displayed in Table 4. The key difference in the results is the ratio of high-

Table 4. Comparison of Reaction-based Slicing and RECAP Slicing Rules[a]

preprocessing method	model	number of compounds found	yield	average mean score in *resulting data set*	ratio of fully satisfied reaction filters
reaction-based slicing	QSAR	10454	0.817	0.729	0.892
	ROCS	10326	0.807	0.597	0.890
RECAP slicing rules	QSAR	8388	0.655	0.506	0.154
	ROCS	8339	0.651	0.462	0.000

[a]We note that the relatively low scores for the ROCS component arise from the fact that it is more difficult to satisfy the query completely; unlike for a QSAR model, scores nearing 1 are not expected.

scoring molecules capable of satisfying the imposed RFs. Whereas the model trained on data sliced according to reaction rules consistently scores very highly and therefore produces libraries synthesizable *via* these two reactions, the model trained on data preprocessed using RECAP rules struggles to fulfill these criteria. With the exception of one run, the model fails to learn to follow the reaction routes. This gives clear evidence of the positive effect of the reaction-based slicing method for applications involving specific chemical reactions.

It can be further noted that the ROCS task appears to be more difficult for the models to learn, which is not surprising because the RL agent is asked to learn 3D features with only a 2D representation of its input. This has also been observed in previous work by Grebner *et al.*,[53] where ROCS was used as a 3D similarity scoring component for an RL-based molecular deep generative model. As expected, both the yield and the ratios of

compounds satisfying the RFs are not significantly affected by the change between QSAR and ROCS scoring components; the difference lies in the average scores achieved by compounds in the *resulting data set*. As the training plots in Figure 7 show, this is due to the fact that the scores start relatively low and gradually increase throughout the runs as the agents learn to satisfy the ROCS input.

To understand the diversity of the compounds proposed by agents trained with these two different scoring function components, we further contrast the molecular properties of the decorations proposed by the respective methods when trained on a data set obtained using reaction-based slicing. Figure 8 demonstrates that the change in a scoring function component guiding the training affects the proposed decorations. In the example of attachment point 2, we see that whereas the groups proposed by an agent trained using ROCS are generally lighter, they tend to contain more rings and have more hydrogen-bond acceptors. In some cases, we can further note that the RF has not been satisfied for a share of the output, for example, in the cases where the Buchwald reaction fails to propose a compound containing an aromatic ring. This demonstrates the need for a careful consideration of the scoring function design along with the RFs to achieve the optimal results for a given task. The sample compounds proposed by each of the methods are plotted in Figure 9 for comparison. We can observe the formation of the amide bonds, as required by the RFs, and the previously noted tendency of the ROCS-guided model to propose decorations with multiple rings.

The experiments discussed in this section demonstrate the benefit of reaction-based slicing over the more traditional RECAP rules. This approach to preprocessing implicitly teaches the decorator model to follow chemical reactions and thus increases the probability of learning to satisfy a RF. Figure 7 demonstrates that whereas it is possible for a model trained on compounds sliced using RECAP rules to learn to satisfy a RF, the likelihood of this happening is much lower. The models trained using reaction-based slicing consistently fulfill RFs and other user-defined criteria.

Following Specific Reactions. Using the optimal learning strategy and the prior model pretrained on data sliced using reaction rules, we propose a new set of experiments to demonstrate the effect of selective RFs on the produced libraries. For each of the attachment points, we select a relevant plausible chemical reaction that can serve to introduce desirable moieties. Specifically, sulphonamide coupling is used as an alternative to amide coupling for attachment point 1, and the Buchwald reaction of attachment point 2 may be replaced by a nucleophilic heteroaromatic substitution (S_NAr). We experiment with each of the four possible combinations of these RFs to demonstrate the effect of these filters on the produced compounds. For illustrative purposes, a high-scoring compound discovered for each of these combinations of RFs by a QSAR-guided predictive model is plotted in Figure 10. The RFs have a clear effect on the proposed molecules, enforcing the formations of appropriate bonds.

All of the sets of RFs are applied to the two different setups of the scoring function, as in the previous experiments, using either the QSAR predictive model or the ROCS similarity component to direct the model toward the target chemical subspace of compounds active in the DRD2 data set. The reason for using different scoring functions in this experiment is to demonstrate the effect the scoring function has on the output and decouple
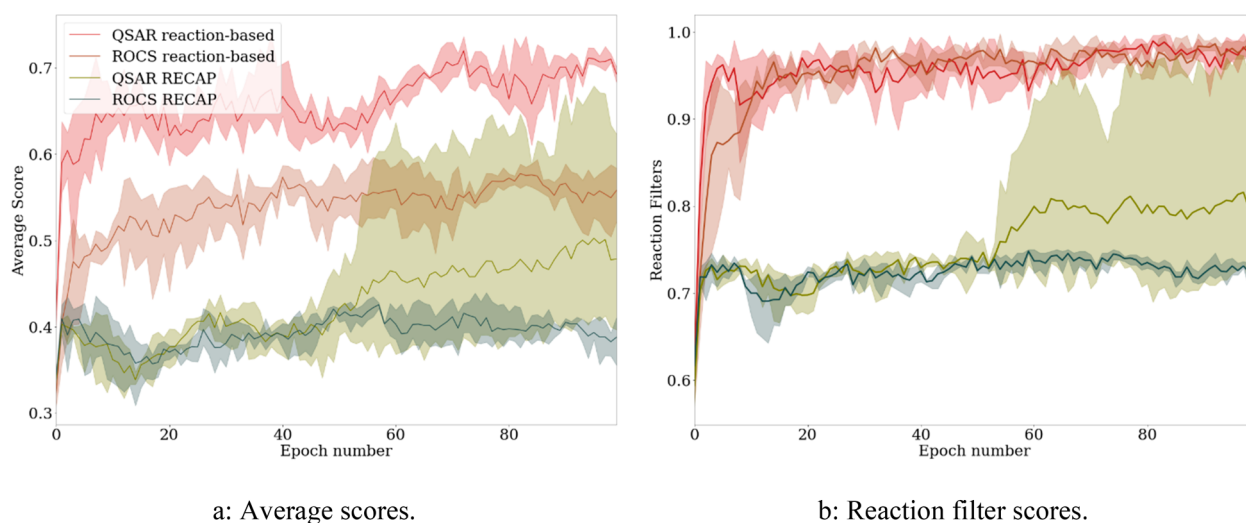
a: Average scores.   b: Reaction filter scores.

Figure 7. Comparison of the two preprocessing approaches. For both QSAR- and ROCS-guided learning, the model trained using reaction-based slicing is consistently superior in terms of the overall score and, moreover, satisfies both reaction filters.
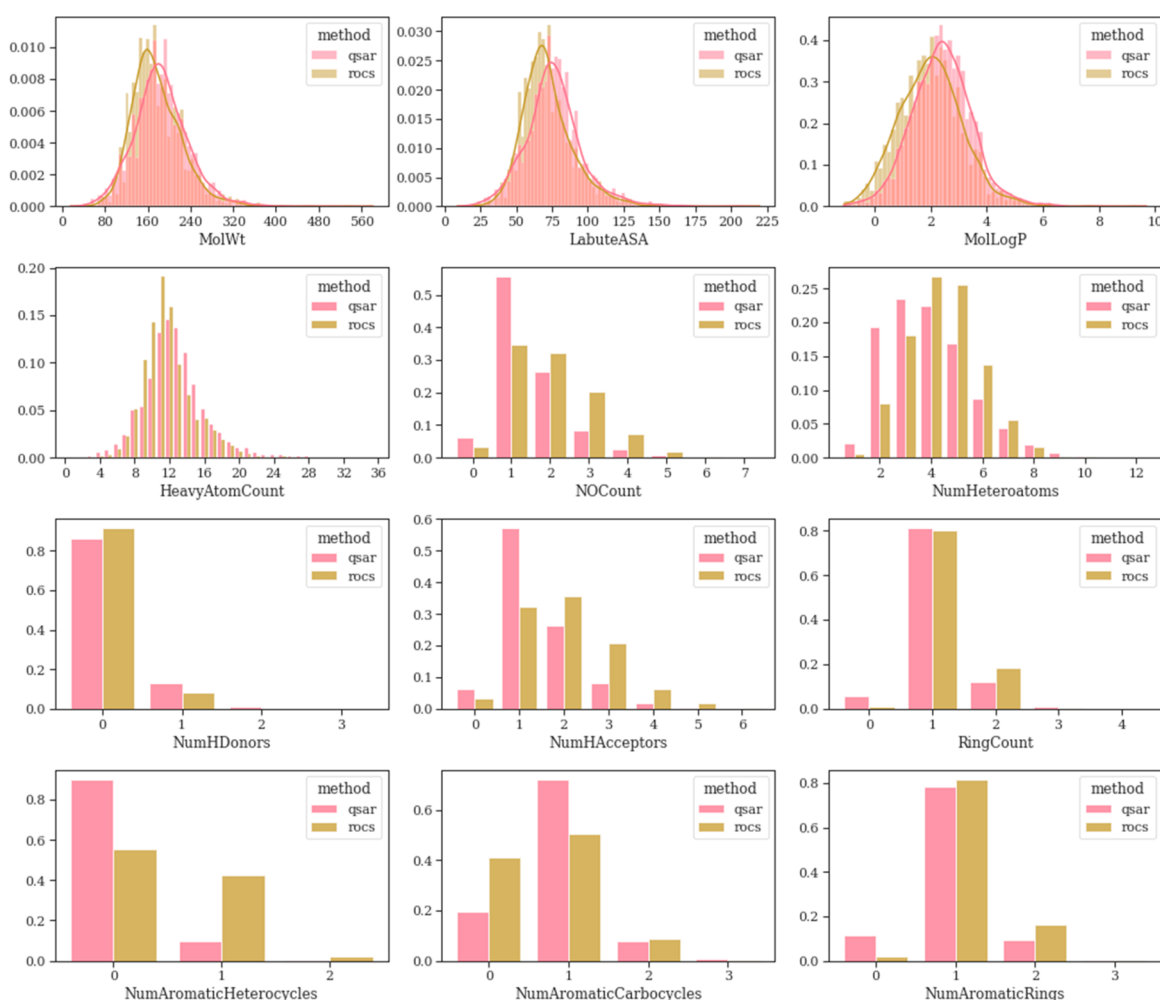

Figure 8. Example molecular properties of decorations for attachment point 2 when the Buchwald reaction filter is imposed.

this from the effect of RFs. The numerical results of these experiments are displayed in Table 5.

The performance of the ROCS model when RFs are exchanged is more stable than that of QSAR, showing similar patterns and an ability to learn to follow different reaction

routes. This is presumably caused by a lower degree of inductive bias built into the model through this scoring component. The consistently lower average scores in the *resulting data set* can be attributed to the greater difficulty of learning this component in general; it is more difficult to score highly the structural
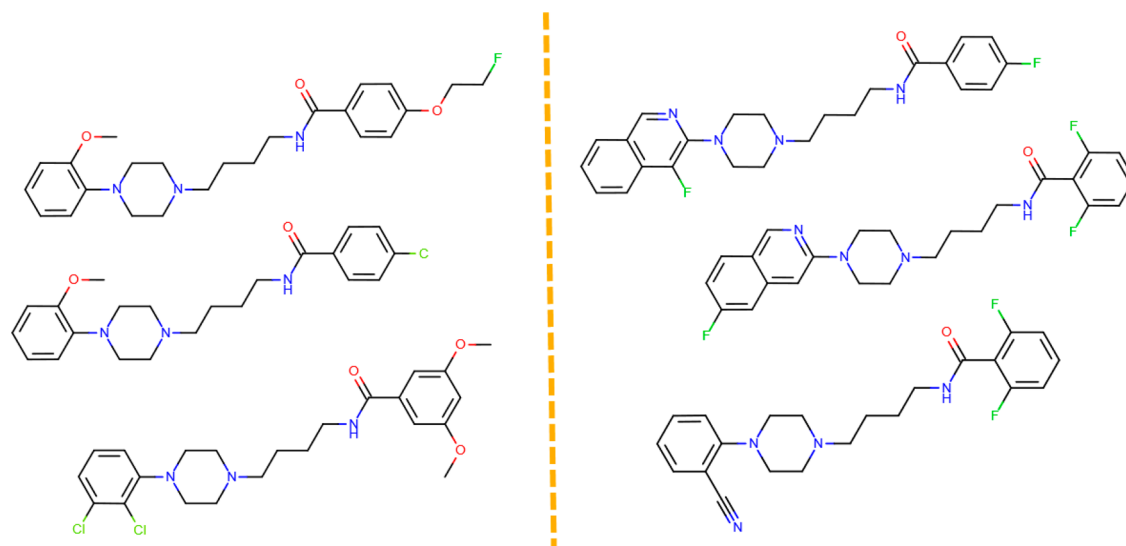
Figure 9. High-scoring compounds proposed by a model guided by a QSAR predictive property (left) and by a ROCS scoring component (right). All of these compounds satisfy both the amide coupling and Buchwald reaction filters.
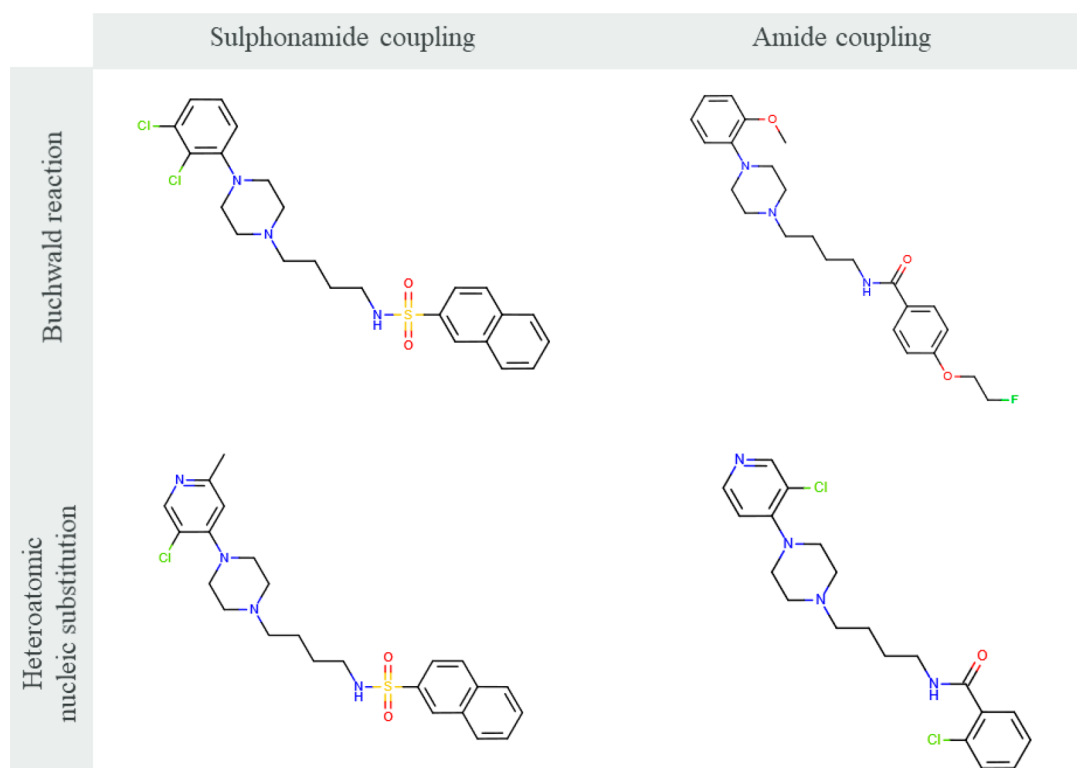

Figure 10. Comparison of compounds proposed by models optimizing for various reaction filters. The formation of the amide and sulphonamide bonds can be seen.

similarity (and match) requirements of a ROCS component. Nevertheless, this does not mean that the model performs badly; on the contrary, the high yields show that it is an effective guide toward a desirable area of the chemical space. Moreover, the highest achievable scores of the ROCS component are typically lower than those for a QSAR model and commonly lie around the value 0.8.

To quantify the differences in the properties of the decorations arising from various RFs, we examine the distributions of the key molecular properties of the proposed functional groups for each attachment point based on the applied RFs. A selection of molecular descriptors of decorations generated for attachment point 1, decorated *via* amide or sulphonamide coupling, is displayed in Figure 11. A significant increase in the weight of the proposed decorations, caused by the presence of more heavy atoms, occurs when the sulphonamide coupling is introduced. Figure 12 further shows selected discrete properties of the decorations proposed for the second attachment point. In both plots, a variation in the distributions can be observed across all four combinations of RFs; the effect of the RFs imposed for the given attachment point is nevertheless clearly notable. This is to be expected

Table 5. Comparison of Varying Reaction Filters for QSAR and ROCS Models[a]

preprocessing method	reaction filter	number of compounds found	yield	average mean score in *resulting data set*	ratio of fully satisfied reaction filters
QSAR model	Buchwald−amide	**10454**	**0.817**	**0.734**	**0.892**
	Buchwald−sulphonamide	10083	0.788	0.688	0.847
	S_NAr−amide	9809	0.766	0.585	0.359
	S_NAr−sulphonamide	9228	0.721	0.641	0.577
ROCS model	Buchwald−amide	**10326**	**0.807**	**0.596**	**0.890**
	Buchwald−sulphonamide	10207	0.797	0.592	0.871
	S_NAr−amide	9837	0.768	0.545	0.551
	S_NAr−sulphonamide	9560	0.747	0.552	0.541

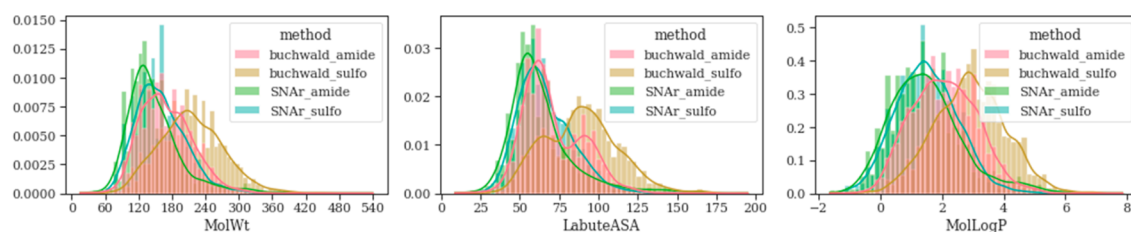[a]Buchwald−amide filter most easily satisfies both models.

Figure 11. Continuous molecular descriptors of decorations of attachment point 1 for each of the four reaction filters applied, trained using a QSAR model. This attachment point is decorated by either amide or sulphonamide coupling.
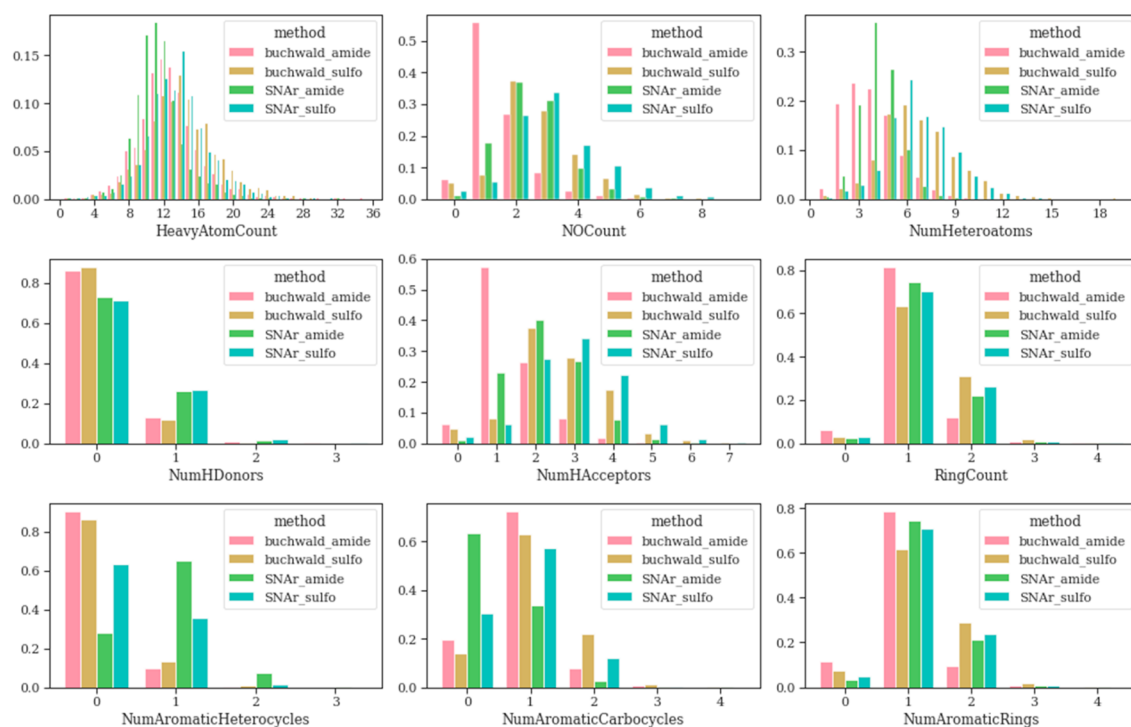

Figure 12. Discrete molecular descriptors of the decorations of attachment point 2 for each of the four reaction filters applied. Note that this attachment point is decorated using either the Buchwald reaction or the S_NAr substitution; the differences observed in this plot therefore primarily arise as a result of this reaction filter.

because the agent receives rewards based on all of the compounds proposed, but each attachment point is strongly influenced by the prescribed reaction.

As a final point of comparison of the RFs, Figure 13 displays the distribution of selected molecular properties for each of the two attachment points using the original RF composed of amide coupling and the Buchwald reaction. In general, amide coupling produces somewhat smaller and lighter decorations. Moreover, the distributions tend to be less peaked and centered around the

mode, which is to be expected for this reaction because it is more general. Once again, we further note that not all proposed compounds satisfy the Buchwald RF because decorations missing an aromatic ring are proposed for attachment point 2.

Scaffolds with Varied Numbers of Attachment Points. So far, all experiments focused on a two-attachment point scaffold. To give a fair picture of the decorator's abilities, we additionally introduce tasks working with scaffolds containing one to four decoration points. Because the purpose of this section is
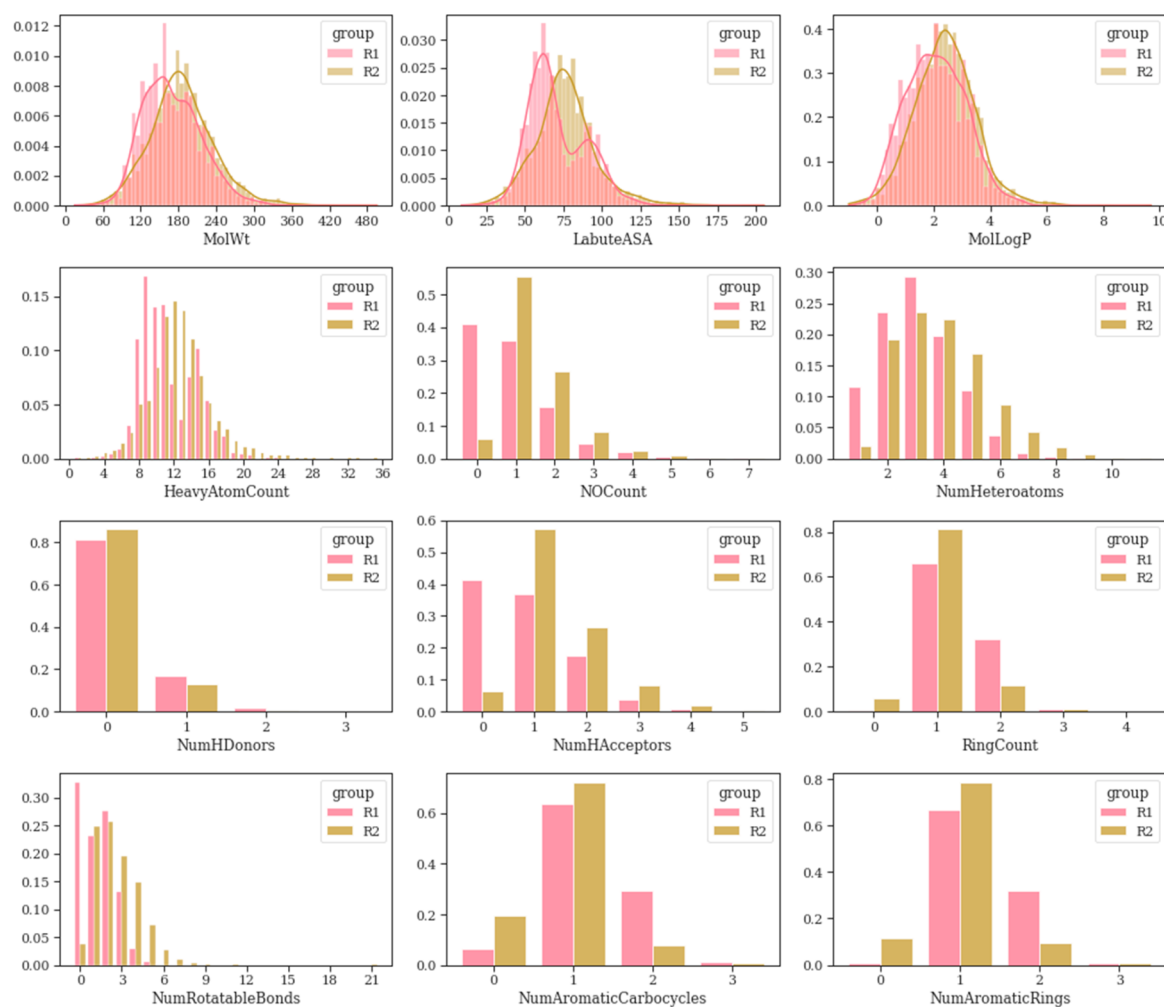
Figure 13. Comparison of the properties of the decorations proposed for each attachment point. The first attachment is decorated using amide coupling; the second attachment is decorated *via* the Buchwald reaction.

primarily proof of concept, we restrict our attention to simple experiments aiming to force the model to start growing large enough decorations to satisfy molecular weight requirements. For simplicity, a RF is not implemented here. The experiments nevertheless demonstrate that the decorator is capable of working with these scaffolds to produce unique and valid compounds.

For the purpose of these simple experiments, we use two scaffolds from the DRD2 data set with one and three attachment points. In both cases, the weight requirement on the final compound is for it to lie between 450 and 650. These values have been chosen to force the original scaffolds to grow significantly without leaving the domain of chemically reasonable compounds in the output because there are no other constraints to guide the model. The scaffolds are displayed in Figure 14.
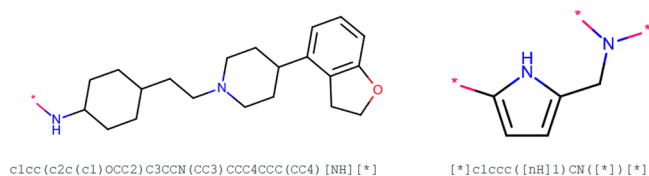

c1cc(c2c(c1)OCC2)C3CCN(CC3)CCC4CCC(CC4)[NH][*]   [*]c1ccc([nH]1)CN([*])[*]

Figure 14. Scaffolds with one and three attachment points used in the final experiment.

As demonstrated in the plots in Figure 15, the model does not struggle with any of these tasks and rapidly adjusts to the requirement and starts to generate compounds in the appropriate molecular weight range. Similarly, the validity of the proposed output is consistently >90%. These experiments clearly show that the use cases of the decorator model include working with scaffolds of varying numbers of attachment points.

■ DISCUSSION

We have designed and executed a range of experiments to establish the abilities of the newly proposed LibINVENT model. Most importantly, the results clearly demonstrate the model's superiority in learning to follow specific chemical reactions, which is achieved by the introduction of a compliant compound slicing strategy. The decorator model has proven to be capable of rapidly designing libraries of molecules that are synthesizable from a given scaffold by following a set of reactions, as defined by the user. This enables fine control over the output and makes the model widely and readily applicable in a multitude of scenarios. This expands the capabilities of the REINVENT family of generative models with the introduction of a dedicated capability to design focused molecular libraries.

The first task was to determine an optimal learning strategy for setting up the RL rewards. Four different strategies have been proposed based on arguments discussed in the literature.
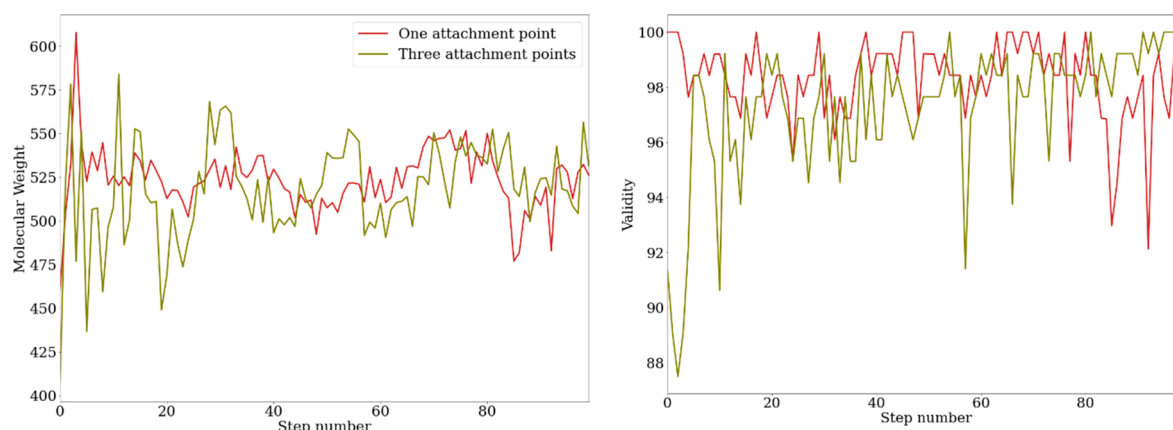
Figure 15. Satisfying weight requirements for varying numbers of attachment points. Both models are capable of proposing valid compounds of required molecular weights.

Whereas it is not immediately intuitive, the DAP learning strategy has proven to be the most successful strategy. The motivation for this reward setup is a "carrot on a stick" scenario. A combination of the prior likelihood and the scoring function is used to guide the agent toward a desirable region of the chemical space while ensuring that the underlying chemical syntax is not forgotten. Two of the remaining strategies, on the contrary, attempted to maximize the score or a sum of the score and the prior likelihood directly. Despite appearing more natural at first glance, this approach does not work as well because the models struggle to retain the ability to propose valid molecules, as they start focusing too much on the score. A possible rationale for this is the notoriously high variance typically observed for policy iteration RL; whereas we note that the generative model requires this variance to explore the chemical space, too much variation combined with a lack of anchor to the prior knowledge is detrimental to the performance. The final method explored in the Article minimizes the square of the loss used for the optimal DAP strategy. This is more mathematically sound because the reward is bounded, but it does not appear beneficial in practice because edge scenarios where the unboundedness of the DAP reward could be an issue rarely arise. We therefore confirm the observations of Olivecrona *et al.* in selecting the DAP strategy as the method of choice.[44]

Two different scoring components have been used to guide the model to propose new ligands for the DRD2 receptor. A simple QSAR property prediction model has the advantage of a faster execution and an overall higher score, but its stronger inductive bias restricts the model to a narrower domain.[54] As a result, a QSAR-based model strongly favors certain decorations and therefore struggles to fulfill some reaction routes incompatible with these functional groups. In the second use case scenario, the ROCS similarity measure was used to demonstrate that various scoring function components may be used to guide the model to a desirable chemical space. A certain degree of experimentation or user intuition is often required to determine the optimal combination of guidance for the model and freedom to explore to obtain the best possible libraries, as each of the scoring components introduces its own biases and benefits. The results nevertheless confirm that LibINVENT is a flexible tool that admits a wide range of inputs and is able to return appropriate output.

An important note regarding the selective RF is that the user is responsible for providing correct and valid reactions for correct attachment points to get a good result. Whereas a range of

reaction definitions is provided in the public repository, the reactions prescribed to a given attachment point have to be feasible. RFs enable the user to specify a variable number of reaction definitions for each attachment point. If an infeasible reaction is required, then the model is not going to be able to fulfill this RF and will always receive a score of 0. Depending on the setup, this can lead to a failed run with a very small and irrelevant output, as the low scores do not guide the agent in the right direction. It is therefore essential that the user is aware of this potential pitfall. A consistently low RF score, plateauing at a value lower than one, is often an indicator of a wrong reaction requirement.

Normally, the idea selection process does not stop with simply generating the final data set. Because the resulting output can contain thousands of entries, it is a common practice to apply additional postprocessing steps to narrow it down to the most relevant molecules. Although we do not demonstrate it here, there are many additional profiling calculations (various ADMET properties and physics-based calculations) that have not been included to the RL run due to performance or accuracy considerations. These are normally applied afterward to help the library selection. Once the right candidates for synthesis are identified, the building blocks can be queried either for in-house availability or for adequate retrosynthetic suggestions by tools such as AiZynthFinder.[55]

Because the current work is focused on the development of the generative method and on the general guidelines for its optimal application, we did not consider conducting a further downstream postprocessing and comprehensive evaluation of its output. However, we do provide an analysis of the combination of one specific scaffold (Figure 3), RFs (Buchwald and amide coupling), and two different scoring components (QSAR and ROCS scoring) in the Supporting Information (in "Comparison with Other Methods"). This analysis is based on metrics loosely inspired by the screening library guidelines from Bayer.[56] The example shows novelty measured as the percentage of generated molecules that are not accessible from commercially available building blocks (aryl halides and carboxylic acids in our example). Both scoring components result in the generation of novel R1, R2 substitutions of the scaffold (ROCS: 15.9%, QSAR: 28.6%, Supporting Information, Table 5). It is obvious that this metric is dependent on the database of commercially available building blocks. Here we used the eMolecules database.[57] We also compare our generative approach with a traditional enumerative approach that would require the

construction of a virtual library with all combinations of available carboxylic acids and aryl halides and the provision of substituents for the given scaffold and subsequent filtration of the virtual molecules with the same scoring functions (either QSAR or ROCS) used in LibINVENT. We show that even the first step of the construction of the virtual library is computationally infeasible because its size is of the magnitude of 10^{12} virtual molecules. The subsequent step of evaluating each and every compound from such a library, especially with the ROCS scoring function, is even more computationally demanding. In contrast, LibINVENT does fewer than 2×10^4 evaluations in total for the provided examples and identifies good scoring molecules for >70% of the evaluations made (yield shown in Table 5). In other words, brute force enumeration may lead to a quite extensive evaluation and thus become rather impractical. Also, generative methods are shown to have a broader chemical space coverage,[58] thus providing the potential to cover solutions that are not reachable through enumeration.

It is important to stress that the *resulting data set* is primarily a product of the input scaffold, the scoring function, the generative model, and the learning algorithm. Given that we propose an optimal learning algorithm and a working generative model, the user has the full freedom (and responsibility) to tailor the learning objectives and combine them with input scaffolds that are of relevance to concrete projects.

CONCLUSIONS

To achieve an efficient and natural symbiosis between computational and traditional wet lab methods in drug discovery, it is essential to overcome a few prevailing bottlenecks. One of the key issues is the low efficiency of incorporating deep learning into the production pipeline caused by complicated lead synthesis and an overly diverse output from generative models. The objective of this work has been to provide a method that can help bridge this gap between *in silico* and *in vitro* drug design by developing a tool that takes the needs of real-life synthesis into consideration and increasing the productivity by reducing the number of DMTA cycles performed.[59]

In this work, we have introduced a flexible tool capable of proposing optimal decorations given a scaffold and a set of user-specified objectives. Thanks to the custom chemical reaction definitions, we can also include in the objectives RFs. LibINVENT therefore enables the rapid generation of focused virtual chemical libraries that can be used for lead optimization and are readily synthesizable *in vitro*. Even when these filters are not specified, the output of the model still benefits from the high synthetic accessibility. The design of the RL loop further introduces a rapid way to focus the model to a desirable part of the chemical space. As the experiments demonstrate, the learning is instantaneous and results in the design of varied and focused chemical libraries.

LibINVENT is a deep-learning-based tool capable of following specific reaction constraints in the design of entire chemical libraries within which the diversity is narrowly focused to a domain determined by the user. This makes it readily applicable in a broad range of scenarios. The model is released in our public repository along with the corresponding code.

DATA AND SOFTWARE AVAILABILITY

The training data set used for pretraining the prior model is available in our public GitHub repository at https://github.com/MolecularAI/Lib-INVENT/. It includes both the purged data set resulting from the removal of non-drug-like compounds from the ChEMBL Database, version 27[18] and the sliced data set obtained by slicing the purged data according to the handcrafted reaction SMIRKS.

The reaction SMIRKS and the software necessary for performing the purging and slicing are available in the LibINVENT data set repository at https://github.com/MolecularAI/Lib-INVENT-dataset. This repository moreover contains tutorials, structured as Jupyter notebooks, detailing the workflow of the data preparation process.

The LibINVENT repository holds all of the code necessary for the training of the decorator model. The pretrained models used to perform the experiments reported in this Article are provided, along with the predictive DRD2 model and the ROCS inputs. Using these, the results reported here can be replicated by following the tutorials included in the repository.

ROCS[52] is a proprietary licensed software released by OpenEye. Version 2019.10.2 of the openeye-toolkits package was used for all of the experiments reported in the Article.

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.1c00469.

> Technical and implementation details such as the hyperparameters used in training and the exact model vocabulary. A breakdown of the results of the reruns of the experiments and a more complete mathematical background of the project (PDF)

AUTHOR INFORMATION

Corresponding Author

Atanas Patronov — *Molecular AI, Discovery Sciences, R&D, AstraZeneca, Gothenburg 43183, Sweden;* orcid.org/0000-0002-9797-6573; Email: atanas.patronov@astrazeneca.com

Authors

Vendy Fialková — *Molecular AI, Discovery Sciences, R&D, AstraZeneca, Gothenburg 43183, Sweden*

Jiaxi Zhao — *Molecular AI, Discovery Sciences, R&D, AstraZeneca, Gothenburg 43183, Sweden; Department of Pharmaceutical Biosciences, Uppsala University, Uppsala 75237, Sweden*

Kostas Papadopoulos — *Molecular AI, Discovery Sciences, R&D, AstraZeneca, Gothenburg 43183, Sweden*

Ola Engkvist — *Molecular AI, Discovery Sciences, R&D, AstraZeneca, Gothenburg 43183, Sweden; Department of Computer Science and Engineering, Chalmers University of Technology, Gothenburg 41756, Sweden;* orcid.org/0000-0003-4970-6461

Esben Jannik Bjerrum — *Molecular AI, Discovery Sciences, R&D, AstraZeneca, Gothenburg 43183, Sweden;* orcid.org/0000-0003-1614-7376

Thierry Kogej — *Molecular AI, Discovery Sciences, R&D, AstraZeneca, Gothenburg 43183, Sweden*

Complete contact information is available at:
https://pubs.acs.org/10.1021/acs.jcim.1c00469

Notes

The authors declare no competing financial interest.

■ REFERENCES

(1) Jiménez-Luna, J.; Grisoni, F.; Schneider, G. Drug Discovery with Explainable Artificial Intelligence. *Nat. Mach. Intell.* **2020**, *2*, 573−584.

(2) Paul, D.; Sanap, G.; Shenoy, S.; Kalyane, D.; Kalia, K.; Tekade, R. K. Artificial Intelligence in Drug Discovery and Development. *Drug Discovery Today* **2021**, *26*, 80−93.

(3) Bush, J. T.; Pogany, P.; Pickett, S. D.; Barker, M.; Baxter, A.; Campos, S.; Cooper, A. W. J.; Hirst, D.; Inglis, G.; Nadin, A.; Patel, V. K.; Poole, D.; Pritchard, J.; Washio, Y.; White, G.; Green, D. V. S. A Turing Test for Molecular Generators. *J. Med. Chem.* **2020**, *63*, 11964−11971.

(4) Blaschke, T.; Olivecrona, M.; Engkvist, O.; Bajorath, J.; Chen, H. Application of Generative Autoencoder in *De Novo* Molecular Design. *Mol. Inf.* **2018**, *37*, 1700123.

(5) Xue, D.; Gong, Y.; Yang, Z.; Chuai, G.; Qu, S.; Shen, A.; Yu, J.; Liu, Q. Advances and Challenges in Deep Generative Models for de Novo Molecule Generation. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **2019**, *9*, No. e1395.

(6) Winter, R.; Montanari, F.; Steffen, A.; Briem, H.; Noé, F.; Clevert, D. A. Efficient Multi-Objective Molecular Optimization in a Continuous Latent Space. *Chem. Sci.* **2019**, *10*, 8016−8024.

(7) Yang, Y.; Zheng, S.; Su, S.; Zhao, C.; Xu, J.; Chen, H. SyntaLinker: Automatic Fragment Linking with Deep Conditional Transformer Neural Networks. *Chem. Sci.* **2020**, *11*, 8312−8322.

(8) Grebner, C.; Matter, H.; Plowright, A. T.; Hessler, G. Automated de Novo Design in Medicinal Chemistry: Which Types of Chemistry Does a Generative Neural Network Learn? *J. Med. Chem.* **2020**, *63*, 8809−8823.

(9) Hughes, J. P.; Rees, S. S.; Kalindjian, S. B.; Philpott, K. L. Principles of Early Drug Discovery. *Br. J. Pharmacol.* **2011**, *162*, 1239−1249.

(10) Langdon, S. R.; Ertl, P.; Brown, N. Bioisosteric Replacement and Scaffold Hopping in Lead Generation and Optimization. *Mol. Inf.* **2010**, *29*, 366−385.

(11) Maziarz, K.; Jackson-Flux, H.; Cameron, P.; Sirockin, F.; Schneider, N.; Stiefl, N.; Brockschmidt, M. Learning to Extend Molecular Scaffolds with Structural Motifs. *arXiv*, March 5, 2021, 2103.03864, ver. 2. https://arxiv.org/pdf/2103.03864.pdf, (accessed 2021-03-18).

(12) Hussain, J.; Rea, C. Computationally Efficient Algorithm to Identify Matched Molecular Pairs (MMPs) in Large Data Sets. *J. Chem. Inf. Model.* **2010**, *50*, 339−348.

(13) Blaschke, T.; Arús-Pous, J.; Chen, H.; Margreitter, C.; Tyrchan, C.; Engkvist, O.; Papadopoulos, K.; Patronov, A. REINVENT 2.0: An AI Tool for De Novo Drug Design. *J. Chem. Inf. Model.* **2020**, *60*, 5918.

(14) Popova, M.; Isayev, O.; Tropsha, A. Deep Reinforcement Learning for de Novo Drug Design. *Sci. Adv.* **2018**, *4*, No. eaap7885.

(15) Mercado, R.; Rastemo, T.; Lindelöf, E.; Klambauer, G.; Engkvist, O.; Chen, H.; Jannik Bjerrum, E. Graph Networks for Molecular Design. *Mach. Learn. Sci. Technol.* **2021**, *2* (2), 025023.

(16) Arús-Pous, J.; Patronov, A.; Bjerrum, E. J.; Tyrchan, C.; Reymond, J. L.; Chen, H.; Engkvist, O. SMILES-Based Deep Generative Scaffold Decorator for de-Novo Drug Design. *J. Cheminf.* **2020**, *12*, 38.

(17) Langevin, M.; Minoux, H.; Levesque, M.; Bianciotto, M. Scaffold-Constrained Molecular Generation. *J. Chem. Inf. Model.* **2020**, *60*, 5637.

(18) Mendez, D.; Gaulton, A.; Bento, A. P.; Chambers, J.; De Veij, M.; Félix, E.; Magariños, M. P.; Mosquera, J. F.; Mutowo, P.; Nowotka, M.; Gordillo-Marañón, M.; Hunter, F.; Junco, L.; Mugumbate, G.; Rodriguez-Lopez, M.; Atkinson, F.; Bosc, N.; Radoux, C. J.; Segura-Cabrera, A.; Hersey, A.; Leach, A. R. ChEMBL: Towards Direct Deposition of Bioassay Data. *Nucleic Acids Res.* **2019**, *47*, D930−D940.

(19) Göller, A. H.; Kuhnke, L.; Montanari, F.; Bonin, A.; Schneckener, S.; ter Laak, A.; Wichard, J.; Lobell, M.; Hillisch, A. Bayer's in Silico ADMET Platform: A Journey of Machine Learning over the Past Two Decades. *Drug Discovery Today* **2020**, *25*, 1702−1709.

(20) Li, Y.; Hu, J.; Wang, Y.; Zhou, J.; Zhang, L.; Liu, Z. DeepScaffold: A Comprehensive Tool for Scaffold-Based De Novo Drug Discovery Using Deep Learning. *J. Chem. Inf. Model.* **2020**, *60*, 77.

(21) Lewell, X. Q.; Judd, D. B.; Watson, S. P.; Hann, M. M. RECAPRetrosynthetic Combinatorial Analysis Procedure: A Powerful New Technique for Identifying Privileged Molecular Fragments with Useful Applications in Combinatorial Chemistry. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 511−522.

(22) Heikamp, K.; Zuccotto, F.; Kiczun, M.; Ray, P.; Gilbert, I. H. Exhaustive Sampling of the Fragment Space Associated to a Molecule Leading to the Generation of Conserved Fragments. *Chem. Biol. Drug Des.* **2018**, *91*, 655−667.

(23) Bradshaw, J.; Kusner, M. J.; Paige, B.; Benevolentai, M. H. S. S.; Miguel Hernández-Lobato, J. Generating Molecules via Chemical Reactions. In *Proceedings of the Seventh International Conference on Learning Representations (ICLR 2019)*, New Orleans, LA, 2019.

(24) Horwood, J.; Noutahi, E. Molecular Design in Synthetically Accessible Chemical Space via Deep Reinforcement Learning. *ACS Omega* **2020**, *5*, 32984.

(25) Arús-Pous, J.; Patronov, A.; Bjerrum, E. J.; Tyrchan, C.; Reymond, J. L.; Chen, H.; Engkvist, O. SMILES-Based Deep Generative Scaffold Decorator for de-Novo Drug Design. *J. Cheminf.* **2020**, *12*, 38.

(26) Kolen, J. F.; Kremer, S. C. *A Field Guide to Dynamical Recurrent Networks* **2009**, 200−203.

(27) Arús-Pous, J.; Johansson, S. V.; Prykhodko, O.; Bjerrum, E. J.; Tyrchan, C.; Reymond, J. L.; Chen, H.; Engkvist, O. Randomized SMILES Strings Improve the Quality of Molecular Generative Models. *J. Cheminf.* **2019**, *11*, 11.

(28) Krishnan, S. R.; Bung, N.; Bulusu, G.; Roy, A. Accelerating *De Novo* Drug Design against Novel Proteins Using Deep Learning. *J. Chem. Inf. Model.* **2021**, *61*, 621.

(29) Chen, H.; Engkvist, O.; Wang, Y.; Olivecrona, M.; Blaschke, T. The Rise of Deep Learning in Drug Discovery. *Drug Discovery Today* **2018**, *23*, 1241−1250.

(30) Olivecrona, M.; Blaschke, T.; Engkvist, O.; Chen, H. Molecular De-Novo Design through Deep Reinforcement Learning. *J. Cheminf.* **2017**, *9*, 9.

(31) Bender, A.; Cortés-Ciriano, I. Artificial Intelligence in Drug Discovery: What Is Realistic, What Are Illusions? Part 1: Ways to Make an Impact, and Why We Are Not There Yet. *Drug Discovery Today* **2021**, *26*, 511−524.

(32) He, J.; You, H.; Sandström, E.; Nittinger, E.; Bjerrum, E. J.; Tyrchan, C.; Czechtizky, W.; Engkvist, O. Molecular Optimization by Capturing Chemist's Intuition Using Deep Neural Networks. *J. Cheminf.* **2021**, *13*, 1−17.

(33) Rifaioglu, A. S.; Nalbat, E.; Atalay, V.; Martin, M. J.; Cetin-Atalay, R.; Doğan, T. DEEPScreen: High Performance Drug-Target Interaction Prediction with Convolutional Neural Networks Using 2-D Structural Compound Representations. *Chem. Sci.* **2020**, *11*, 2531−2557.

(34) Bemis, G. W.; Murcko, M. A. The Properties of Known Drugs. 1. Molecular Frameworks. *J. Med. Chem.* **1996**, *39*, 2887−2893.

(35) Blaschke, T.; Engkvist, O.; Bajorath, J.; Chen, H. *J. Cheminf.* **2020**, *12*, 68.

(36) Kotsias, P.-C.; Arús-Pous, J.; Chen, H.; Engkvist, O.; Tyrchan, C.; Bjerrum, E. J. Direct Steering of de Novo Molecular Generation with Descriptor Conditional Recurrent Neural Networks. *Nat. Mach. Intell.* **2020**, *2*, 254−265.

(37) Li, Y.; Hu, J.; Wang, Y.; Zhou, J.; Zhang, L.; Liu, Z. DeepScaffold: A Comprehensive Tool for Scaffold-Based De Novo Drug Discovery Using Deep Learning. *J. Chem. Inf. Model.* **2020**, *60*, 77.

(38) Thomas, M.; Smith, R. T.; O'Boyle, N. M.; de Graaf, C.; Bender, A. Comparison of Structure- and Ligand-Based Scoring Functions for

Deep Generative Models: A GPCR Case Study. *J. Cheminf.* **2021**, *13*, 1−20.

(39) Sun, J.; Jeliazkova, N.; Chupakhin, V.; Golib-Dzib, J.-F.; Engkvist, O.; Carlsson, L.; Wegner, J.; Ceulemans, H.; Georgiev, I.; Jeliazkov, V.; Kochev, N.; Ashby, T. J.; Chen, H. ExCAPE-DB: An Integrated Large Scale Dataset Facilitating Big Data Analysis in Chemogenomics. *J. Cheminf.* **2017**, *9*, 17.

(40) Bjerrum, E. J. SMILES Enumeration as Data Augmentation for Neural Network Modeling of Molecules. *arXiv*, March 21, 2017, 1703.07076, ver. 2. https://arxiv.org/pdf/1703.07076.pdf, (accessed 2021-04-16).

(41) Arús-Pous, J.; Johansson, S. V.; Prykhodko, O.; Bjerrum, E. J.; Tyrchan, C.; Reymond, J. L.; Chen, H.; Engkvist, O. Randomized SMILES Strings Improve the Quality of Molecular Generative Models. *J. Cheminf.* **2019**, *11*, 71.

(42) Goyal, A.; Lamb, A.; Zhang, Y.; Zhang, S.; Courville, A.; Bengio, Y. Professor Forcing: A New Algorithm for Training Recurrent Networks. In *Advances in Neural Information Processing Systems*; Neural Information Processing Systems Foundation, 2016; pp 4608−4616.

(43) Skinnider, M. A.; Stacey, R. G.; Wishart, D. S.; Foster, L. J. Deep Generative Models Enable Navigation in Sparsely Populated Chemical Space. *ChemRxiv* **2021**, DOI: 10.26434/CHEMRXIV.13638347.V1.

(44) Olivecrona, M.; Blaschke, T.; Engkvist, O.; Chen, H. Molecular De-Novo Design through Deep Reinforcement Learning. *J. Cheminf.* **2017**, *9*, 48.

(45) Williams, R. J. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Mach. Learn.* **1992**, *8*, 229−256.

(46) Wu, C.; Rajeswaran, A.; Duan, Y.; Kumar, V.; Bayen, A. M.; Kakade, S.; Mordatch, I.; Abbeel, P. Variance Reduction for Policy Gradient with Action-Dependent Factorized Baselines. *arXiv*, March 20, 2018, 1803.07246, ver. 1. https://arxiv.org/pdf/1803.07246.pdf, (accessed 2021-03-23).

(47) Morris, G. M.; Lim-Wilby, M. Molecular Docking. *Methods Mol. Biol.* **2008**, *443*, 365−382.

(48) Kumar, A.; Zhang, K. Y. J. Advances in the Development of Shape Similarity Methods and Their Application in Drug Discovery. *Front. Chem.* **2018**, *6*, 315.

(49) Korshunova, M.; Huang, N.; Capuzzi, S.; Radchenko, D. S.; Savych, O.; Moroz, Y. S.; Wells, C.; Willson, T. M.; Tropsha, A.; Isayev, O. A Bag of Tricks for Automated De Novo Design of Molecules with the Desired Properties: Application to EGFR Inhibitor Discovery. *ChemRxiv* **2021**, DOI: 10.26434/chemrxiv.14045072.v1.

(50) Landrum, G. *RDKit*, 2021. http://www.rdkit.org/ (accessed 2020-02-12).

(51) de Souza Neto, L. R.; Moreira-Filho, J. T.; Neves, B. J.; Maidana, R. L. B. R.; Guimarães, A. C. R.; Furnham, N.; Andrade, C. H.; Silva, F. P. In Silico Strategies to Support Fragment-to-Lead Optimization in Drug Discovery. *Front. Chem.* **2020**, *8*, 93.

(52) *ROCS 3.4.1.2*; OpenEye Scientific Software: Santa Fe, NM, 2021. http://www.eyesopen.com, (accessed 2021-04-28).

(53) Grebner, C.; Matter, H.; Plowright, A. T.; Hessler, G. Automated de Novo Design in Medicinal Chemistry: Which Types of Chemistry Does a Generative Neural Network Learn? *J. Med. Chem.* **2020**, *63*, 8809−8823.

(54) Baxter, J. A Model of Inductive Bias Learning. *J. Artif. Intell. Res.* **2000**, *12*, 149.

(55) Genheden, S.; Thakkar, A.; Chadimová, V.; Reymond, J. L.; Engkvist, O.; Bjerrum, E. AiZynthFinder: A Fast, Robust and Flexible Open-Source Software for Retrosynthetic Planning. *J. Cheminf.* **2020**, *12*, 1−9.

(56) Follmann, M.; Briem, H.; Steinmeyer, A.; Hillisch, A.; Schmitt, M. H.; Haning, H.; Meier, H. An Approach towards Enhancement of a Screening Library: The Next Generation Library Initiative (NGLI) at Bayer — against All Odds? *Drug Discovery Today* **2019**, *24* (3), 668−672.

(57) Index of /free/2021-07-01. https://downloads.emolecules.com/free/2021-07-01/ (accessed 2021-07-09).

(58) Arús-Pous, J.; Blaschke, T.; Ulander, S.; Reymond, J. L.; Chen, H.; Engkvist, O. Exploring the GDB-13 Chemical Space Using Deep Generative Models. *J. Cheminf.* **2019**, *11*, 20.

(59) Schneider, P.; Walters, W. P.; Plowright, A. T.; Sieroka, N.; Listgarten, J.; Goodnow, R. A.; Fisher, J.; Jansen, J. M.; Duca, J. S.; Rush, T. S.; Zentgraf, M.; Hill, J. E.; Krutoholow, E.; Kohler, M.; Blaney, J.; Funatsu, K.; Luebkemann, C.; Schneider, G. Rethinking Drug Design in the Artificial Intelligence Era. *Nat. Rev. Drug Discovery* **2020**, *19*, 353−364.

R