

## **DrugSynthMC: an atom based generation of drug-like molecules with Monte Carlo Search**

Milo Roucairol<sup>1</sup>, Alexios Georgiou<sup>1</sup>, Tristan Cazenave<sup>1\*</sup>, Filippo Prischi<sup>2\*</sup>, Olivier E. Pardo<sup>3\*</sup>.

<sup>1</sup> LAMSADE, Université Paris-Dauphine, Pl. du Maréchal de Lattre de Tassigny, 75016 Paris, France

<sup>2</sup> Randall Centre for Cell and Molecular Biophysics, School of Basic and Medical Biosciences, King's College London, London SE1 1UL, United Kingdom.

<sup>3</sup> Division of Cancer, Department of Surgery and Cancer, Imperial College, Du Cane Road, London W12 0NN, United Kingdom

## Abstract

A growing number of Deep Learning (DL) methodologies have recently been developed to design novel compounds and expand the chemical space within virtual libraries. Most of these Neural Network approaches design molecules to specifically bind a target, based on its structural information and/or knowledge of previously identified binders. Fewer attempts have been made to develop approaches for de novo design of virtual libraries, as synthesizability of generated molecules remains a challenge. In this work, we developed a new Monte Carlo Search (MCS) algorithm, DrugSynthMC (Drug Synthetise using Monte Carlo), in conjunction with DL and statistical-based priors to generate thousands of interpretable chemical structures and novel drug-like molecules per second. DrugSynthMC produces drug-like compounds using an atom-based search model that builds molecules as SMILES, character by character. Designed molecules follow Lipinski's "rule of 5", show a high proportion of predicted-to-be synthesisable compounds and efficiently expand the chemical space within the libraries, without reliance on training datasets, synthesizability metrics or enforcing during SMILES generation. Our approach can function with or without an underlying Neural Network and is thus easily explainable and versatile. This ease in drug-like molecule generation allows for future integration of score functions aimed at different target- or job -oriented goals. Thus, DrugSynthMC is expected to enable the functional assessment of large compound libraries covering an extensive novel chemical space, overcoming the limitations of existing drug collections. The software is available at <https://github.com/RoucairoIMilo/DrugSynthMC>

## Introduction

Since the 1980s, *in silico* approaches have been extensively and routinely used in drug discovery and have transformed the medicinal chemistry field <sup>1</sup>, with expectation to do so even more in the future. The need for rapid response, highlighted by the emergence of resistant bacteria and, among others, the COVID-19 pandemic, has fuelled the development of novel computational tools for drug design and screening <sup>2</sup>. *In silico* virtual-library screening (VS) is usually the first critical step in structure-based drug discovery, where the algorithm aims to predict the best matching binding mode of a ligand to a receptor <sup>3</sup>. Despite the many attempts to improve accuracy of VS methods <sup>4,5</sup>, the relatively limited chemical diversity of compounds in libraries reduces the ability of structure-based VS to identify hits and leads <sup>6,7</sup>. Indeed, it has been estimated that only a small portion ( $10^6$ - $10^7$ ) of the  $10^{63}$  drug-like molecules predicted to be synthetically accessible has been explored <sup>8</sup>.

Several studies have shown that screening larger libraries that expand the accessible molecules by several order of magnitude ( $\sim 10^{11}$ ) improves the rate of true high affinity (nM-pM) binders <sup>9-12</sup>. To further expand the chemical space within virtual libraries, generative models based on Deep Learning (DL) methodologies have been used to produce molecules with desired chemical features able to bind specifically macromolecules of interest extensively reviewed in <sup>13, 14-16</sup>.

Recurrent neural networks (RNNs) were among the first DL methods to be developed to generate SMILES, a line notation that describes the structure of a molecule <sup>17</sup>. However, RNNs tend to suffer from exposure bias, and a diverse range of alternative approaches that differ in the training procedure and model architecture have been proposed. These include Variational AutoEncoders (VAEs) <sup>18</sup>, Generative Adversarial Networks (GANs) <sup>19</sup>, and graph-based generator <sup>20</sup>. Nevertheless, even these alternative approaches have limitations and, for example, it has been reported that VAEs generated SMILES often cannot be translated into interpretable chemical structures <sup>21</sup>. Furthermore, a key requirement of generative models is that designed molecules must be synthesisable. A wide range of different approaches have been used to predict synthetic feasibility of molecules, including using scores based on structure complexity and similarity to evaluate synthesizability <sup>22</sup>, or integrating computer aided synthesis planning (CASP) tools as part of the design process <sup>23, 24</sup>. However, as Bilodeau et al. <sup>25</sup> recently highlighted, approaches that embed CASP tools automatically inherit CASP limitations, thus reducing chemical diversity of compounds in de novo generated libraries.

In this paper, we use Monte Carlo Search (MCS) algorithms in conjunction with DL and statistical-based priors to generate thousands of interpretable chemical structures and novel drug-like molecules per second. DrugSynthMC (Drug Synthetise using Monte Carlo) relies on an algorithm never previously used in chemistry/medicinal chemistry, differing from prior efforts in that it rapidly produces valid molecules, while being explainable and, importantly, requiring no training. The algorithm does not enforce synthesizability or rely on synthesizability metrics during SMILES generation, which has been shown to reduce diversity and novelty of generated compounds <sup>26</sup>. However, the synthesizability of generated compounds was analysed using an open-source retrosynthesis analysis tool, AiZynthFinder <sup>27</sup>. We show that our method generates drug-like libraries with a high proportion of predicted-to-be synthesisable compounds and efficiently expands the chemical space within the libraries. Finally, DrugSynthMC is highly flexible and, in future, could be easily tuned using multiple parameters to tailor for a wide range of different chemical goals and/or create customised libraries of compounds for specific targets.

## Methods

### Search model

The search model consists of a set of instructions defining what available moves (character addition) can be applied to unfinished SMILES. All SMILES start empty, and only one atom can be added at that stage. Once atoms are added, cycles and subtrees can be initiated. This search model ensures that initiated SMILES can be completed from any point of the search into a valid final SMILE, with generation being heavily restricted by the below rules. The central rule is to respect the total number of bonds each atom can form. To do so, the number of available bonds of the last added atom in the

current subtree is stored. This number is checked to compute possible legal moves and decreased, when adding a character corresponding to a new atom in a different subtree level, according to the number of bonds used to connect the new atom to the previous one. To maximise the chance of generating valid molecules with drug-like properties, we scanned the FDA subset of the ZINC20 database<sup>28</sup> to obtain frequency information on types of atoms and bonds involved. This information was stored in frequency matrices, allowing to label as illegal moves by the search model bonds that were never or very rarely encountered (less than 1/10000 of the bonds for each atom involved). It also enabled us to focus the generator on using the most commonly encountered atoms only: carbon (C), oxygen (O), nitrogen (N), fluorine (F), sulphur (S), and chlorine (Cl), while bromine (Br) and phosphate (P) atoms were not included due to their relative rarity, but could be easily reintroduced in the search model, along with inorganic atoms. Last, we added shortcuts for the different bonding of S: bond to 2 atoms, or 4 atoms in sulfinyl, or 6 atoms in sulfonyl. Rather than learning the entire functional groups and edge possibilities through the prior, we decided to pre-process the SMILES prior to training, turning the trifluoromethyl (W), sulfinyl (M), and sulfonyl (U) into their own symbols used in building SMILES.

### Operating principle

To give an accurate list of possible characters to append to an incomplete SMILE, the search model keeps track of several parameters concerning the SMILES at any step:

(1) The depths of the nested subtrees: The subtrees are expressed in SMILES language using the "(" symbol for opening, and the ")" symbol for closure. Termination of SMILES is not allowed unless they are back to the root tree, meaning that all open parentheses must be closed. Closing a subtree when no parenthesis is open is also forbidden.

(2) Covalence bounds counts: Each subtree contains an active atom, which is the last one added at that level. It is the atom to which other atoms, cycles, and subtrees are then added. The search model keeps track of the number of available covalent bounds on the last atom of each subtree until the latter is closed by a ")". Moves that exceed the number of covalent bounds available are not allowed.

(3) Cycle nesting: Nested cycles that share more than one bond are uncommon in drug-like molecules. Thus only the most recent cycle is allowed to be terminated.

(4) Cycle length: More rules, not inherent to the SMILES grammar, were added to improve the drug-likeness and stability of the molecules. These included not allowing cycles smaller than 5 and bigger than 7 atoms to be generated. Indeed, while these cycle sizes exist in drug-like molecules, size 4 cycles are usually hard to synthesize and unstable, while cycles longer than 7 atoms are rare.

Additionally, to avoid unnecessarily long playouts and molecules, a boolean flag called "finish ASAP" is added. It is set to true once a certain number of characters is met and disallows certain moves, such as opening a new subtree or cycle, with some exceptions (i.e., finishing an already open subtree). This complicated set of rules is necessary to prevent the search from cornering itself due to cycles or using all the covalent bounds available on a particular level. However, this results in a

rather lengthy function enumerating the legal moves from an incomplete SMILES (about 100 lines long).

### Playouts

In Monte Carlo Search, a playout is a computationally cheap unfolding of actions from a starting search space state. Moves are selected and played until the resulting new search is terminal or no move is available. The terminal state is then evaluated and returned to be used by the algorithm calling the playout. Playouts are a core element of most Monte Carlo Search algorithms and the move selection process differentiates algorithms. Many algorithms use random playouts, usually when tackling NP-hard problems where expert knowledge can't help. The Prior Upper Confidence bounds applied to Trees (PUCT)<sup>29</sup> used in DeepMind's AlphaGo<sup>30</sup> employs a neural network to recommend moves to play in a game of Go. The Nested Rollout Policy Adaptation (NRPA)<sup>29</sup> learns a reinforcement learning policy on the fly to select the moves. Some other applications can evaluate non-terminal states and use greedy playouts (our unpublished results). In fact, the choice of playout mode can be more important to the success of a MCS than the algorithm used (our unpublished results). In this study, we used the Sampling method to act as a baseline to compare our algorithms. This method consists of independent playouts from the start state and ends once a molecule reaches the best possible score.

### Guided playouts: ngrams

Ngrams are short subsequences derived from larger sequences of characters. They were one of the first machine learning approaches, mostly used in Natural Language Processing (NLP). Through the use of grams, it is possible to compute the statistics of every sequence of characters in a learning corpus to predict the next character in a Markovian process. Here, ngrams were generated through extracting every sequence of characters from the FDA subset of the ZINC20 database and were associated conditional probabilities used to guide the playout of an incomplete SMILES given its last characters at any step. For instance, if the learning corpus only contained "COCC" and "COCO", the entry for "COC" would report  $P(C|COC) = 0.5$  and  $P(O|COC) = 0.5$ . The ngrams were only used to value moves and act as a prior, following the rules of the search model. Hence, whenever the ngram valued a move that was forbidden by the model, the move was discarded. The ngrams also used the cycle length computed from the FDA-approved compound database to guaranty the right proportion of cycles of each length in the final molecules. Storing information about the cycles' length within the ngrams themselves would be possible but expected to be more error-prone. Therefore, we instead decided to use a probability to end a cycle at a certain length. These probabilities are 0.228 for a cycle of length 5, 0.751 for a cycle of length 6, and 0.02 for a cycle of length 7. Smaller and longer cycles were omitted for reasons mentioned above.

### Guided playouts: neural network

As previously explored<sup>31</sup>, neural networks can be trained to give conditional probabilities of the next character given all of the SMILES. The neural network is then repeated on the newly extended SMILES until the SMILES is terminal, similar to recent Large Language Models (LLM). This method strays from the usual playouts as it is not limited by the rules of the search model, but is prone to produce invalid smiles<sup>31</sup>. The same neural network can also be used as a prior, following the rules of the search model just like in the case of ngrams. This method has two advantages over approaches using ngrams: (1) taking the entire context of the SMILES into account, and (2) generalization (as ngrams require an exact precursor). However, ngrams are faster, more easily explainable, and require no training.

The neural network used in the present study is the same as in<sup>31</sup>, and was trained using the same dataset. It was only slightly modified to accommodate for the shortcuts with no apparent repercussions on the results obtained.

### Monte Carlo Search: Upper Confidence bounds applied to Trees

Monte Carlo Search (MCS) encompasses a wide range of search algorithms. These differ from regular search algorithms in that they are not deterministic and use randomness to explore search spaces too large for regular algorithms and learn guiding policies<sup>32</sup>. In 2006, Kocsis introduced a variant of Monte Carlo Tree Search (MCTS) called Upper Confidence bounds applied to Trees (UCT) or Bandit based Monte-Carlo Planning<sup>33</sup>. It is now the most widely used MCTS and MCS algorithm in the literature, often in the form of Prior UCT (PUCT).

UCT is a bandit-based reinforcement learning algorithm similar to Q-learning. The algorithm learns a policy and selects to go down the tree in order to balance exploitation and exploration.

Like all MCTS, UCT is comprised of 4 phases:

1. Selection: Progress in the selection tree according to the policy.
2. Expansion: Once a state that has not been explored yet is encountered, it is added to the tree.
3. Evaluation: A playout (or another fast algorithm) is used to evaluate the quality of the new state.
4. Backpropagation: The result of the evaluation is used to update all the parent states visited during the Selection step.

These four steps are repeated indefinitely, starting from the initial state each time, much like in Q-learning. What differs between the various MCTS algorithms is the formula of the policy.

UCT uses the following formula to evaluate a child state S to select:

$$UCT_s = X_s + C * \sqrt{\frac{\ln V}{V_s}}$$

Where  $X_s$  is the average score of state S, C is the exploration/exploitation constant (usually 1), V is the number of visits of the current state, and  $V_s$  is the number of child state S visits.

PUCT is a generalization of UCT. It uses a prior to guide not only the playouts but also the selection process, allowing to speed up the latter with knowledge from outside this execution. PUCT uses a different selection formula:

$$UCTs = X_s + C * P_s * \frac{\sqrt{V}}{1 + V_s}$$

With  $P_s$  the value given by the prior for state  $S$ .

### Nested Monte Carlo Search

Nested Monte Carlo Search (NMCS) is a different type of MCS algorithm. It uses lower-level NMCS on each available move of the current state to explore the search tree and register the sequences of actions leading to the best scores. Once the lower-level NMCS returns their best routes to the higher level, it executes the next move of the best route and calls new lower-level NMCS on the resulting state. The lowest level NMCS is (usually) a playout. Unlike UCT, the NMCS does not explore the entire search space given enough time, but gains in precision as it explores the tree and is less prone to be stuck in a local maximum. This property led to generally better results from NMCS over UCT and other algorithms on optimization problems.

### Validation of the Generated Small

An *in silico* validation was performed to understand the quality and synthesizability of the generated molecules. SMILES validity and clogP were calculated with the RDKit<sup>34</sup>. The AiZynthFinder software was used to predict synthetic routes of generated SMILES<sup>27</sup>.

## **Results and Discussion**

### **Performance of Models**

The aim of this work was to (i) generate large sets of novel molecules that expand the chemical diversity of available compounds libraries, (ii) generate small molecules with drug-like properties, (iii) which in future could be used for VS campaigns and could be easily grown into larger drugs and tailored for specific targets. As such, DrugSynthMC generates, in the absence of any training, SMILES of drug-like molecules without prior targets in mind. Thus, the score function is meant to only maximize the validity and drug-likeness of the output molecules and is not goal-oriented (e.g., tailored to bind a specific biological target). The drug-likeness is obtained through maximizing the general chemical properties associated with drugs according to the “Rule of 5”<sup>35-37</sup>. Indeed, easily calculated physicochemical descriptors, such as molecular weight (MW) and number of hydrogen bond donors and acceptors, have been found to correlate with the success rate of clinical trials<sup>38</sup>.

The function of compliance (score function) was defined as:

$$\alpha_1 = -\max(\text{mass} - 500)/500 \quad (1)$$

$$\alpha_2 = -\max(\text{natoms} - 70)/70 \quad (2)$$

$$\alpha_3 = \min(\text{natoms} - 20)/20 \quad (3)$$

$$\alpha_4 = -\max(\text{nhbd} - 5)/5 \quad (4)$$

$$\alpha_5 = -\max(\text{nhba} - 10)/10 \quad (5)$$

$$\text{score} = \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5 \quad (6)$$



With *mass* being the MW of the molecule, *natoms* the number of atoms (including hydrogens), *nhbd* the number of hydrogen bond donors, and *nhba* the number of hydrogen bond acceptors. This formula has the advantage of being computationally cheap and requiring only a pass through the SMILES string.

To identify the most efficient method, we compared the UCT and NMCS MCS algorithms combined with different types of playouts. (i) **random**: the next character is selected uniformly randomly among the ones proposed by the search model. (ii) **enforced**: as in random, the next character is selected uniformly randomly among the ones proposed by the search model. In order to generate compounds that are structurally valid and synthetically accessible, the score function aims to generate molecules containing the same heavy atoms (C, O, N, F, S) ratios as in FDA drugs (retrieved from the FDA subset of the ZINC20 database)<sup>28</sup>. (iii) **ngram**: the next character in the SMILES is selected randomly among the ones proposed by the search model, according to the conditional probabilities of the 3 characters ngram computed on the SMILES from the FDA subset. To balance the number of rings containing 5, 6, and 7 atoms, the score function uses the probability of difference rings occurrence calculated on FDA drugs to end a ring at a certain length. No conditional probability was used to balance type of ring (i.e., aromatic and aliphatic, homo- and hetero-cycles). Additionally, characters with a probability under 0.001 are pruned as they are judged too rare. (iv) **neural**: the next character is selected randomly among the ones proposed by the search model, according to the neural network output weight based on the incomplete SMILES input. As for ngram, characters with a probability under 0.001 are pruned. In addition, we used the Sampling method as a control. This method consists of independent playouts from the start state, and ends once a molecule reaches the best score possible.

For NMCS, we used a level of 3. For PUCT/UCT we used a constant of 1. We use PUCT instead of UCT when a prior is used (ngrams and neural). Here, DrugSynthMC was run on Rust 1.59, on an Intel Core i7-11850H 2.50GHz using a single core, to generate 1000 valid drug-like molecules in independent runs (Table S1). Generation times are not dependent on molecules size or complexity. In all cases, the neural playout is much slower (approximately 5000 times) than the ngrams. The random and enforced playouts do not use a policy and show how the algorithm selection can affect the generation speed. PUCT and UCT use a timeout of 10 sec, because both methods lock into local maxima induced by the shortcuts, failing to generate molecules (unless restarted immediately, turning them into Sampling). Indeed, shortcuts in PUCT and UCT increase the size of molecules and the linked scores, but often produce molecules which deviate from desirable drug-like properties<sup>35-37</sup>. In contrast, UCT without shortcuts is able to return molecules for random and enforced generation. However, while UCT without shortcuts has identical generation times as NMCS and Sampling with



random playouts, with enforced playouts UCT is over 100 and 6 times slower than NMCS and Sampling, respectively.

The NMCS shows a clear advantage over UCT and Sampling when a more complex score function is used. The design of NMCS forces it to explore other subtrees of the search tree, thus preventing locking in local maxima. However, this feature increases generation times when using a prior in this set of experiments. Nevertheless, as suggested by the random and enforced playouts generation times, NMCS is likely to outperform UCT and Sampling for specific goal-oriented generation requiring more complex score functions.

### **Validation of the Generated Drug-like Molecules**

In recent years, with the expansion of DL methods for drug design, several initiatives have been launched to assess generated compounds, which includes benchmarks such as Guacamol and MOSES<sup>39,40</sup>. However, these benchmarks are not suitable for methods like DrugSynthMC that don't exclusively rely on training datasets. Instead, we evaluated similar metrics (validity, uniqueness, novelty, diversity, physicochemical properties and synthesizability) and used comparable tools (RDkit, ZINC databases, AiZynthFinder) to validate our algorithm.

**Table 1:** Validity of 10,000 SMILES

	<b>random</b>	<b>enforced</b>	<b>ngram</b>	<b>neural</b>
<b>NMCS</b>	100%	100%	100%	100%
<b>PUCT</b>	-	-	100%	100%
<b>sampling</b>	100%	100%	100%	100%

To assess the reliability of the tool to generate valid and interpretable molecules, 10,000 generated SMILES were translated into structure representations using RDkit<sup>34</sup>. In all cases, we see that RDkit is able to read and translate 100% of the inputted SMILES (Table 1), showing no syntax errors. This is significantly superior to other methods, which showed validity scores ranging from 85% for generative autoencoders to about 96% for RNN-based models<sup>41-43</sup>.

**Table 2:** Novelty of 10,000 SMILES

	<b>random</b>	<b>enforced</b>	<b>ngram</b>	<b>neural</b>
<b>NMCS</b>	100.00%	99.99%	99.98%	99.96%
<b>PUCT</b>	-	-	99.99%	100.00%
<b>sampling</b>	100.00%	100.00%	100.00%	99.99%

**Table 3:** Uniqueness of 10,000 SMILES

	random	enforced	ngram	neural
<b>NMCS</b>	99.94%	99.17%	81.78%	87.12%
<b>PUCT</b>	-	-	86%	85.32%
<b>sampling</b>	99.83%	98%	82.58%	68.01%

The ability to generate novel compounds was determined by measuring the percentage of molecules in a library of 10,000 generated SMILES which was not present within ZINC-250K<sup>44</sup> (containing nearly 250,000 molecules) (Table 2). In all cases, we see a high level of novelty. It is logical to assume that designing compounds based on general physicochemical properties of drugs instead of on a training set allows DL methods to explore a wider chemical space. Within each of the libraries generated, uniqueness was assessed by identifying the number of identical molecules (Table 3). This shows substantial differences among the different playouts used, with ngram and neural showing a higher number of replicated molecules within libraries. This is linked to the priors restricting the search space to what is more probable. Indeed, by determining the average edit distance as a measure of structural similarity of compounds (Table 4), we can see that both priors similarly restrict the explored chemical space (lower values indicating more similar structure, more likely to have similar properties<sup>45</sup>). Conversely, by generating larger drugs with SMILES string containing 30 characters or more the uniqueness rises above 95% with all methods. While possible, it is more challenging to predict synthesizability of larger drugs using retrosynthesis programmes which search a synthetic route for a chosen molecule<sup>46, 47</sup>.

**Table 4:** Average Distance for of 1,000 SMILES compared 2 by 2

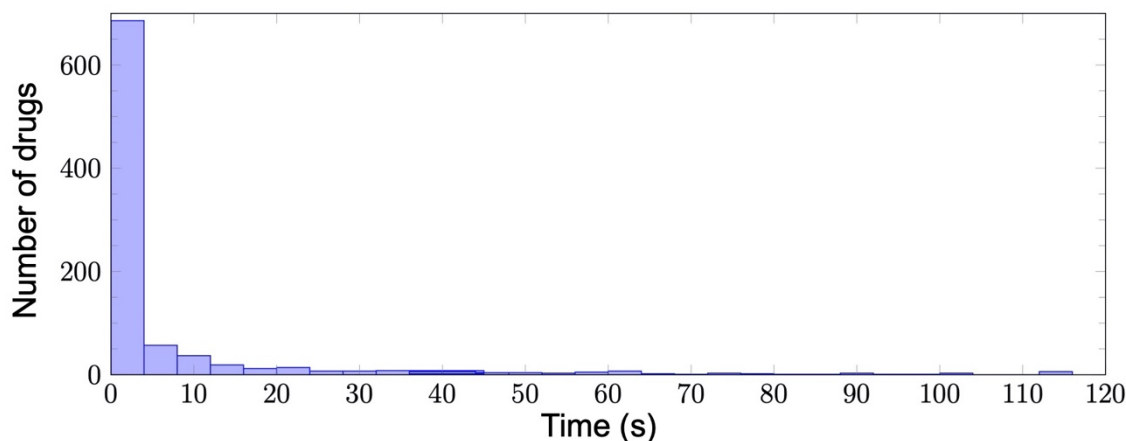
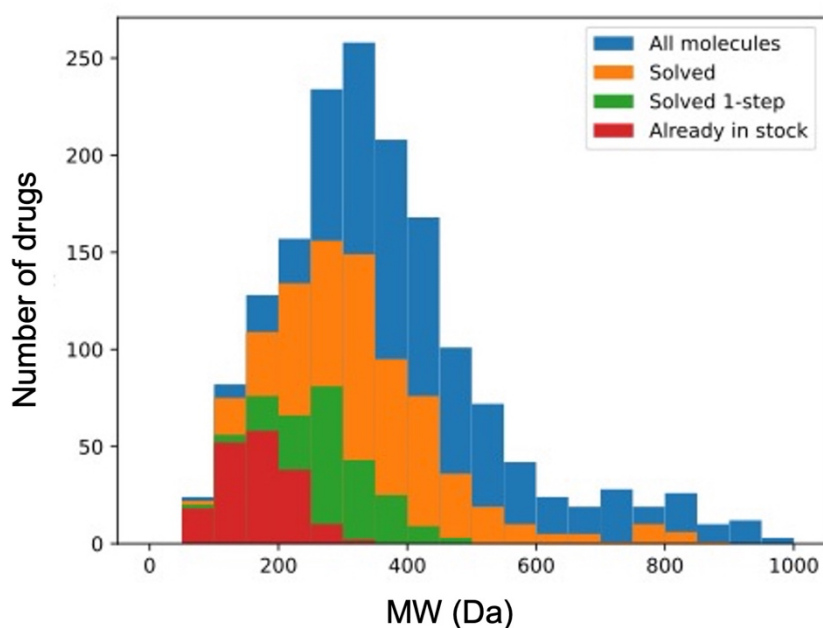
	random	enforced	ngram	neural
<b>NMCS</b>	17.93	14.266	12.954	13.988
<b>PUCT</b>	-	-	13.334	13.956
<b>sampling</b>	17.801	13.94	13.1	14.051

### **Synthesizability of the Generated Drug-like Molecules**

A recent comparison of tools used to predict synthesizability of compounds carried out by Sanchez-Garcia et al.<sup>48</sup> showed that retrosynthesis programmes tend to be more accurate than SA scores.

Yet effectiveness and efficiency of different retrosynthesis tools varies quite dramatically and is still restricted in the variety of reaction types considered <sup>49</sup>. Retrosynthesis programmes use a combination of search algorithms and 1- or 2-step retrosynthesis deep learning to apply reactions to a molecule and divide it into reactants available on the market. AiZynthFinder is an open-source full-fledged template-based retrosynthesis planning framework that adopts a root-parallelized MCTS <sup>27</sup>, similar to ASKCOS <sup>50</sup>, and to our UCT. AiZynthFinder usually tends to successfully identify paths to synthesis in less than 1-2 min <sup>27</sup>. Indeed, this was consistent with our retrosynthesis analysis of drugs retrieved from the FDA subset, showing that AiZynthFinder finds routes for the majority of molecules within the first 2 min of search (Fig. 1A). Hence, 2 min was chosen as maximum search time for the retrosynthesis assessment of the 1000 molecules generated. As mentioned above, there is a connection between number of characters in a SMILES and success in identifying retrosynthesis pathways. We noticed that the larger the compounds generated the less likely AiZynthFinder successfully completes a search within 2 minutes or more. Indeed, AiZynthFinder finds retrosynthesis routes for ~60% of the 1615 FDA-approved drugs and tends to fail for larger molecules (Fig. 1A-B). Therefore, as proof-of-principle study, we chose to generate lower MW drug-like compounds, at the expenses of uniqueness, to show the ability of our approach to generate synthesizable compounds. Due to the similarities between the algorithms when using neural or ngram payouts, and, as shown above, the overall better performances of NMCS, we focused on the molecules generated by this latter method. The ngram payout is the one that generates the largest number of synthesizable compounds. The predicted 32.2% synthesizability is promising and in line with previously reported accuracy rates for different retrosynthesis programmes <sup>49</sup>. The neural payouts, while still promising, provide a much lower rate of predicted synthesizability. This can be explained by the fact that the neural network does not return the exact conditional probability from the training set, and thus rare moves such as the shortcuts are over-represented in these generations. We adopted the neural network from Yang et al. <sup>31</sup> with adaptations for our shortcuts and explicit bonds. While it showed worse outcomes with the implicit bonds and no shortcuts, it delivered similar results with explicit bonds and no shortcuts. However, with more fine tuning, neural networks may have the potential to reach the same level of synthesizability achieved with the ngrams although with the added disadvantage of having a slower execution.

The random and enforced generations act as control experiments. As no policy governs the structure of the generation, nothing can direct the molecule generation toward a sensible and synthesizable outcome. Indeed, AiZynthFinder is unable to propose reactions, thus ending the search long before the 2 minutes time limit (generally in less than a second).

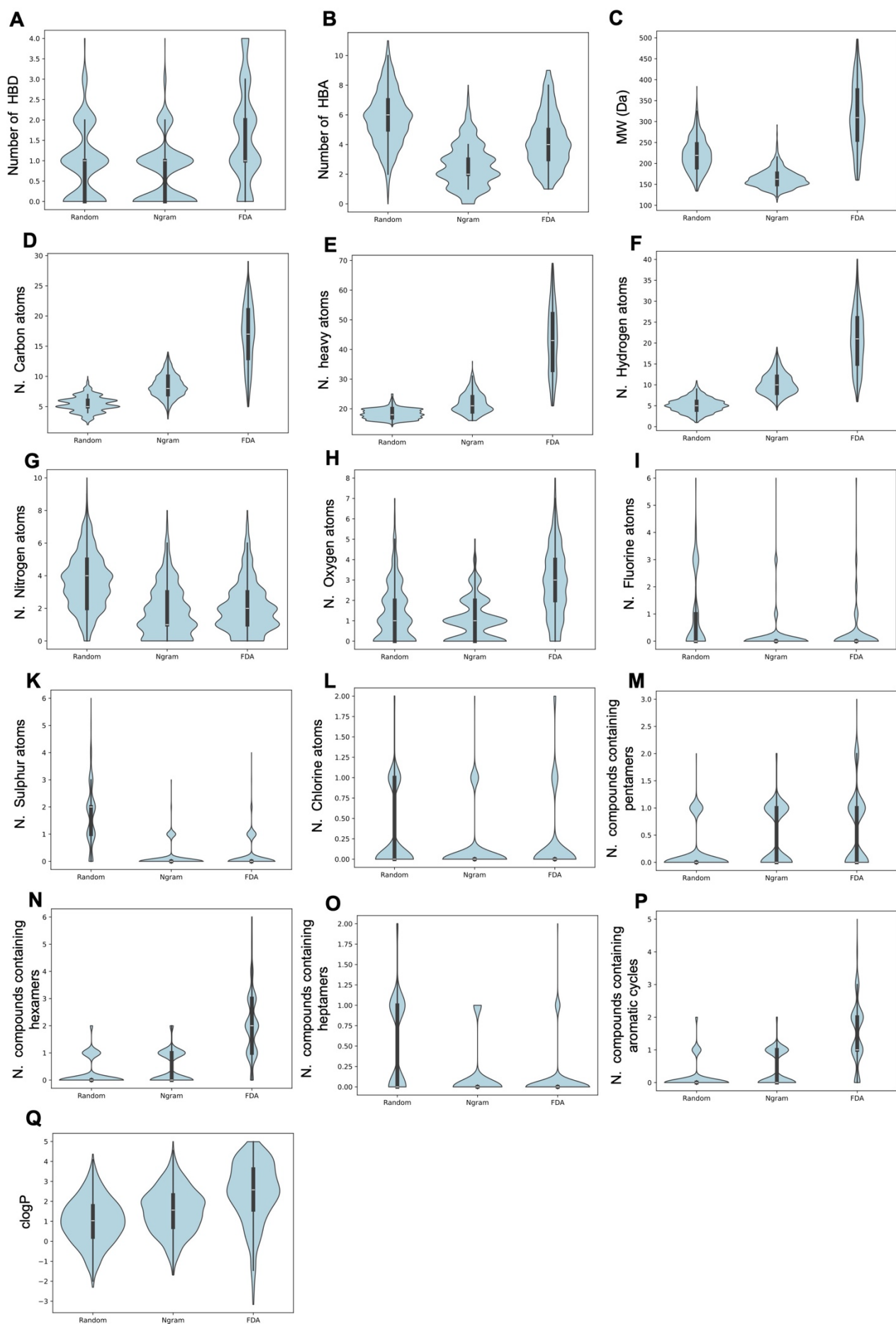
**A****B**

**Figure 1. AiZynthFinder synthetic routes search (A)** histogram plot showing the number of drugs from a dataset comprising 909 drugs retrieved from the FDA subset of the ZINC20 database versus AiZynthFinder synthetic routes search time in seconds. **(B)** histogram plot showing the number drugs for which AiZynthFinder successfully finds synthetic routes in less than 2 minutes versus drugs Molecular Weight. “Solved 1-step” drugs that can be produced directly from commercially available compounds; “Already in stock” drugs that are identified as commercially available by AiZynthFinder.

### **Physicochemical properties of the Generated Drug-like Molecules**

To further validate drug-likeness of generated compounds, we compared distributions of key physicochemical properties of molecules generated with NMCS with ngram payout (the method that generated the highest number of valid and synthesisable drugs-like compounds) with molecules generated with NMCS with random payout (used as control) and drugs retrieved from the FDA subset of ZINC-20<sup>28</sup> abiding to the same general physicochemical properties we took into consideration in the design stage.

It's been shown that, to avoid reducing oral bioavailability, the number of hydrogen bond donors (HBD) should be lower than 6 and hydrogen bond acceptors (HBA) lower than 15<sup>51, 52</sup>. Despite the upper limitation of 10 and 5 for HBA and HBD, respectively, we found that compounds generated with ngram peak at 2 and 1 for HBA and HBD, respectively, with an overall lower number of HBA than random, and a higher ratio of compounds with fewer HBD than random and FDA (Fig. 2A-B). By design, ngram generates compounds with lower MW and total number of carbons (and as a consequence, lower number of heavy atoms) and hydrogens than drugs in FDA (Fig. 2C-F). This is an indispensable requirement for (i) future optimisation studies where compounds may need to be grown to adapt to pockets in targets and increase overall affinity, and (ii) synthesizability analysis with AiZynthFinder. The distribution of heavy atoms is overall comparable in all plots, as the score function generates molecules containing the same heavy atoms (Nitrogen, Oxygen, Fluorine, Sulphur, Chlorine) ratios as in FDA drugs (Fig. 2G-L). DrugSynthMC score function balances the number and size of rings (Fig. 2M-O), but not the type (aromatic and aliphatic, homo- and heterocycles), based on the probability of difference rings occurrence calculated on FDA drugs. The formula returns drugs with a nearly equal distribution of compounds with zero or one aromatic cycle, which is lower than the aromatic cycle content in FDA drugs (Fig. 2P). However, it has been shown that oral drugs with less than 3 aromatic rings have good compound developability<sup>53</sup>, suggesting that DrugSynthMC has the potential to generate compounds with low risk of attrition in early stage development. Generated compounds have also promising oral bioavailability parameters. Earlier work by Caminero Gomes Soares et al.<sup>54</sup> showed that FDA-approved drugs in the last 20 years have a relatively stable numbers of rotatable bonds (mean 5, median 7.5) with about 89% drugs containing less than 10 rotatable bonds. ngrams successfully generates compounds with less than 6 rotatable bonds (mean 3) and, in proportion, produced higher number of molecules with fewer and higher rotatable bonds than the FDA drugs and Random ployout, respectively, thus potentially identifying a balance between flexibility and diffusional cross-section (Fig. 2P).



**Figure 2. Physicochemical properties of the Generated Drug-like Molecules.** Comparison of physicochemical properties among drug-like compounds generated with Random and Ngram, and the "Rule of 5" drugs with the FDA subset of the ZINC20 database. Violin plots showing (A) number of hydrogen bond donors (HBD), (B) hydrogen bond acceptors (HBA), (C) Molecular Weight (MW), (D) total number of Carbons, (E) heavy atoms, (F) Hydrogens, (G) Nitrogen, (H) Oxygen, (I) Fluorine, (K) Sulphur, (L) Chlorine, (M) pentamers, (N) hexamers, (O) heptamers, (P) aromatic cycles, and (Q) clogP. Physicochemical properties were calculated on libraries containing 1,000 generated compounds.

The logarithm of the octanol–water partition coefficient (logP) is a widely used parameter to define solubility of compounds in water and to define their upper limit of intrinsic solubility. Extensive analysis of experimental logP and calculated logP (clogP) of approved drugs over the past 30 years showed that the cutoff <5 remained constant with a mean of about 3.4<sup>55-57</sup>. Unlike other Rule of 5 physicochemical descriptors which are routinely calculated based on SMILES or compounds structures, the clogP is highly dependent on the calculation method, with recent approaches relying on computationally expensive and state-of-the-art neural approaches<sup>58</sup>. Faster approaches are available, but they trade accuracy for speed. As such, to avoid making the evaluation function computationally demanding or inaccurate, the calculation of the logP was not included in the score function and instead clogP was calculated on DrugSynthMC generated compounds using Rdkit, which provides an implementation of the atom-based Wildman-Crippen method<sup>59</sup>. All three sets of molecules analysed have overall similar clogP (Fig. 2Q). The distribution of clogP showed that the majority of molecules have a value between -2 and 4, with ngram showing a larger number of compounds with lower clogP (~2) than that of the FDA, suggesting high likelihood of a compound being orally active.

## Conclusions

Our approach, DrugSynthMC, is capable of designing novel and chemically diverse drug-like compounds by generating character-by-character SMILES. We compared different algorithms with different payouts. Unexpectedly, we showed that the ngrams payout is superior to the more advanced neural approaches and usual MCTS algorithms, such as UCT. Very likely, this is linked to ngrams' ability to follow more closely the distribution. Conversely, as expected, NMCS outperforms UCT in random and enforced generation, as it has been shown before that NMCS is usually better than UCT on optimization problems<sup>60, 61</sup>.

We have shown that DrugSynthMC is highly flexible and, in future, could be easily tuned using multiple parameters. This would result in the generation of customised drugs libraries tailored for specific binding pockets on targets.

## References

1. Zhang, H.; Chen, S., Cyclic peptide drugs approved in the last two decades (2001–2021). *RSC Chemical Biology* **2022**.



2. Zhavoronkov, A., Artificial Intelligence for Drug Discovery, Biomarker Development, and Generation of Novel Chemistry. *Mol Pharm* **2018**, 15, 4311-4313.
3. Salmaso, V.; Moro, S., Bridging Molecular Docking to Molecular Dynamics in Exploring Ligand-Protein Recognition Process: An Overview. *Front Pharmacol* **2018**, 9, 923.
4. Grinter, S. Z.; Zou, X., Challenges, applications, and recent advances of protein-ligand docking in structure-based drug design. *Molecules* **2014**, 19, 10150-76.
5. Yuriev, E.; Holien, J.; Ramsland, P. A., Improvements, trends, and new ideas in molecular docking: 2012-2013 in review. *J Mol Recognit* **2015**, 28, 581-604.
6. Scior, T.; Bender, A.; Tresadern, G.; Medina-Franco, J. L.; Martinez-Mayorga, K.; Langer, T.; Cuanalo-Contreras, K.; Agrafiotis, D. K., Recognizing pitfalls in virtual screening: a critical review. *J Chem Inf Model* **2012**, 52, 867-81.
7. Trezza, A.; Iovinelli, D.; Santucci, A.; Prischi, F.; Spiga, O., An integrated drug repurposing strategy for the rapid identification of potential SARS-CoV-2 viral inhibitors. *Sci Rep* **2020**, 10, 13866.
8. Bohacek, R. S.; McMartin, C.; Guida, W. C., The art and practice of structure-based drug design: a molecular modeling perspective. *Med Res Rev* **1996**, 16, 3-50.
9. Alon, A.; Lyu, J.; Braz, J. M.; Tummino, T. A.; Craik, V.; O'Meara, M. J.; Webb, C. M.; Radchenko, D. S.; Moroz, Y. S.; Huang, X. P.; Liu, Y.; Roth, B. L.; Irwin, J. J.; Basbaum, A. I.; Shoichet, B. K.; Kruse, A. C., Structures of the sigma(2) receptor enable docking for bioactive ligand discovery. *Nature* **2021**, 600, 759-764.
10. Lyu, J.; Wang, S.; Balias, T. E.; Singh, I.; Levit, A.; Moroz, Y. S.; O'Meara, M. J.; Che, T.; Alga, E.; Tolmachova, K.; Tolmachev, A. A.; Shoichet, B. K.; Roth, B. L.; Irwin, J. J., Ultra-large library docking for discovering new chemotypes. *Nature* **2019**, 566, 224-229.
11. Sadybekov, A. A.; Sadybekov, A. V.; Liu, Y.; Iliopoulos-Tsoutsouvas, C.; Huang, X. P.; Pickett, J.; Houser, B.; Patel, N.; Tran, N. K.; Tong, F.; Zvonok, N.; Jain, M. K.; Savych, O.; Radchenko, D. S.; Nikas, S. P.; Petasis, N. A.; Moroz, Y. S.; Roth, B. L.; Makriyannis, A.; Katritch, V., Synthon-based ligand discovery in virtual libraries of over 11 billion compounds. *Nature* **2022**, 601, 452-459.
12. Stein, R. M.; Kang, H. J.; McCorvy, J. D.; Glatfelter, G. C.; Jones, A. J.; Che, T.; Slocum, S.; Huang, X. P.; Savych, O.; Moroz, Y. S.; Stauch, B.; Johansson, L. C.; Cherezov, V.; Kenakin, T.; Irwin, J. J.; Shoichet, B. K.; Roth, B. L.; Dubocovich, M. L., Virtual discovery of melatonin receptor ligands to modulate circadian rhythms. *Nature* **2020**, 579, 609-614.
13. Meyers, J.; Fabian, B.; Brown, N., De novo molecular design and generative models. *Drug Discov Today* **2021**, 26, 2707-2715.
14. Bhisetti, G.; Fang, C., Artificial Intelligence-Enabled De Novo Design of Novel Compounds that Are Synthesizable. *Methods Mol Biol* **2022**, 2390, 409-419.
15. Shiammala, P. N.; Duraimutharasan, N. K. B.; Vaseeharan, B.; Alothaim, A. S.; Al-Malki, E. S.; Snekka, B.; Safi, S. Z.; Singh, S. K.; Velmurugan, D.; Selvaraj, C., Exploring the artificial intelligence and machine learning models in the context of drug design difficulties and future potential for the pharmaceutical sectors. *Methods* **2023**, 219, 82-94.
16. Grisoni, F., Chemical language models for de novo drug design: Challenges and opportunities. *Curr Opin Struct Biol* **2023**, 79, 102527.
17. Weininger, D., SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences* **2002**, 28, 31-36.
18. Gomez-Bombarelli, R.; Wei, J. N.; Duvenaud, D.; Hernandez-Lobato, J. M.; Sanchez-Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Adams, R. P.; Aspuru-Guzik, A., Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Cent Sci* **2018**, 4, 268-276.
19. Putin, E.; Asadulaev, A.; Ivanenkov, Y.; Aladinskiy, V.; Sanchez-Lengeling, B.; Aspuru-Guzik, A.; Zhavoronkov, A., Reinforced Adversarial Neural Computer for de Novo Molecular Design. *J Chem Inf Model* **2018**, 58, 1194-1204.
20. Li, Y.; Zhang, L.; Liu, Z., Multi-objective de novo drug design with conditional graph generative model. *J Cheminform* **2018**, 10, 33.
21. Segler, M. H. S.; Kogej, T.; Tyrchan, C.; Waller, M. P., Generating Focused Molecule Libraries for Drug Discovery with Recurrent Neural Networks. *ACS Cent Sci* **2018**, 4, 120-131.

22. Coley, C. W.; Rogers, L.; Green, W. H.; Jensen, K. F., SCScore: Synthetic Complexity Learned from a Reaction Corpus. *J Chem Inf Model* **2018**, 58, 252-261.
23. Horwood, J.; Noutahi, E., Molecular Design in Synthetically Accessible Chemical Space via Deep Reinforcement Learning. *ACS Omega* **2020**, 5, 32984-32994.
24. Button, A.; Merk, D.; Hiss, J. A.; Schneider, G., Automated de novo molecular design by hybrid machine intelligence and rule-driven chemical synthesis. *Nature Machine Intelligence* **2019**, 1, 307-315.
25. Bilodeau, C.; Jin, W.; Jaakkola, T.; Barzilay, R.; Jensen, K. F., Generative models for molecular discovery: Recent advances and challenges. *WIREs Computational Molecular Science* **2022**, 12, e1608.
26. Fang, Y.; Pan, X.; Shen, H. B., De novo drug design by iterative multiobjective deep reinforcement learning with graph-based molecular quality assessment. *Bioinformatics* **2023**, 39.
27. Genheden, S.; Thakkar, A.; Chadimova, V.; Reymond, J. L.; Engkvist, O.; Bjerrum, E., AiZynthFinder: a fast, robust and flexible open-source software for retrosynthetic planning. *J Cheminform* **2020**, 12, 70.
28. Irwin, J. J.; Tang, K. G.; Young, J.; Dandarchuluun, C.; Wong, B. R.; Khurelbaatar, M.; Moroz, Y. S.; Mayfield, J.; Sayle, R. A., ZINC20-A Free Ultralarge-Scale Chemical Database for Ligand Discovery. *J Chem Inf Model* **2020**, 60, 6065-6073.
29. Rosin, C. D., Multi-armed bandits with episode context. *Annals of Mathematics and Artificial Intelligence* **2011**, 61, 203-230.
30. Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; van den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; Dieleman, S.; Grewe, D.; Nham, J.; Kalchbrenner, N.; Sutskever, I.; Lillicrap, T.; Leach, M.; Kavukcuoglu, K.; Graepel, T.; Hassabis, D., Mastering the game of Go with deep neural networks and tree search. *Nature* **2016**, 529, 484-9.
31. Yang, X.; Zhang, J.; Yoshizoe, K.; Terayama, K.; Tsuda, K., ChemTS: an efficient python library for de novo molecular generation. *Sci Technol Adv Mater* **2017**, 18, 972-976.
32. Metropolis, N.; Ulam, S., The Monte Carlo method. *J Am Stat Assoc* **1949**, 44, 335-41.
33. Kocsis, L.; Szepesvári, C. Bandit Based Monte-Carlo Planning. In *Machine Learning: ECML 2006*, Berlin, Heidelberg, 2006//, 2006; Fürnkranz, J.; Scheffer, T.; Spiliopoulou, M., Eds. Springer Berlin Heidelberg: Berlin, Heidelberg, 2006; pp 282-293.
34. Landrum, G. *rdkit/rdkit: 2023\_09\_6 (Q3 2023) Release*, Zenodo: 2024.
35. Doak, B. C.; Kihlberg, J., Drug discovery beyond the rule of 5 - Opportunities and challenges. *Expert Opin Drug Discov* **2017**, 12, 115-119.
36. Lipinski, C. A., Drug-like properties and the causes of poor solubility and poor permeability. *J Pharmacol Toxicol Methods* **2000**, 44, 235-49.
37. Lipinski, C. A.; Lombardo, F.; Dominy, B. W.; Feeney, P. J., Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Adv Drug Deliv Rev* **2001**, 46, 3-26.
38. Takacs, G.; Sandor, M.; Szalai, Z.; Kiss, R.; Balogh, G. T., Analysis of the uncharted, druglike property space by self-organizing maps. *Mol Divers* **2022**, 26, 2427-2441.
39. Brown, N.; Fiscato, M.; Segler, M. H. S.; Vaucher, A. C., GuacaMol: Benchmarking Models for de Novo Molecular Design. *J Chem Inf Model* **2019**, 59, 1096-1108.
40. Polykovskiy, D.; Zhebrak, A.; Sanchez-Lengeling, B.; Golovanov, S.; Tatanov, O.; Belyaev, S.; Kurbanov, R.; Artamonov, A.; Aladinskiy, V.; Veselov, M.; Kadurin, A.; Johansson, S.; Chen, H.; Nikolenko, S.; Aspuru-Guzik, A.; Zhavoronkov, A., Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models. *Front Pharmacol* **2020**, 11, 565644.
41. Krishnan, S. R.; Bung, N.; Vangala, S. R.; Srinivasan, R.; Bulusu, G.; Roy, A., De Novo Structure-Based Drug Design Using Deep Learning. *J Chem Inf Model* **2022**, 62, 5100-5109.
42. Handa, K.; Thomas, M. C.; Kageyama, M.; Iijima, T.; Bender, A., On the difficulty of validating molecular generative models realistically: a case study on public and proprietary data. *J Cheminform* **2023**, 15, 112.
43. Schoenmaker, L.; Bequignon, O. J. M.; Jespers, W.; van Westen, G. J. P., UnCorrupt SMILES: a novel approach to de novo design. *J Cheminform* **2023**, 15, 22.
44. Akhmetshin, T.; Lin, A. I.; Mazitov, D.; Ziaikin, E.; Madzhidov, T.; Varnek, A., In; 2021.
45. Garcia-Hernandez, C.; Fernandez, A.; Serratos, F., Ligand-Based Virtual Screening Using Graph Edit Distance as Molecular Similarity Measure. *J Chem Inf Model* **2019**, 59, 1410-1421.

46. Engkvist, O.; Norrby, P. O.; Selmi, N.; Lam, Y. H.; Peng, Z.; Sherer, E. C.; Amberg, W.; Erhard, T.; Smyth, L. A., Computational prediction of chemical reactions: current status and outlook. *Drug Discov Today* **2018**, *23*, 1203-1218.
47. Coley, C. W.; Green, W. H.; Jensen, K. F., Machine Learning in Computer-Aided Synthesis Planning. *Acc Chem Res* **2018**, *51*, 1281-1289.
48. Sanchez-Garcia, R.; Havasi, D.; Takács, G.; Robinson, M. C.; Lee, A.; von Delft, F.; Deane, C. M., CoPriNet: graph neural networks provide accurate and rapid compound price prediction for molecule prioritisation. *Digital Discovery* **2023**, *2*, 103-111.
49. Jiang, Y.; Yu, Y.; Kong, M.; Mei, Y.; Yuan, L.; Huang, Z.; Kuang, K.; Wang, Z.; Yao, H.; Zou, J.; Coley, C. W.; Wei, Y., Artificial Intelligence for Retrosynthesis Prediction. *Engineering* **2023**, *25*, 32-50.
50. Coley, C. W.; Thomas, D. A., 3rd; 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.; Jensen, K. F., A robotic platform for flow synthesis of organic compounds informed by AI planning. *Science* **2019**, 365.
51. Doak, B. C.; Zheng, J.; Dobritsch, D.; Kihlberg, J., How Beyond Rule of 5 Drugs and Clinical Candidates Bind to Their Targets. *J Med Chem* **2016**, *59*, 2312-27.
52. Doak, B. C.; Over, B.; Giordanetto, F.; Kihlberg, J., Oral druggable space beyond the rule of 5: insights from drugs and clinical candidates. *Chem Biol* **2014**, *21*, 1115-42.
53. Ritchie, T. J.; Macdonald, S. J., The impact of aromatic ring count on compound developability--are too many aromatic rings a liability in drug design? *Drug Discov Today* **2009**, *14*, 1011-20.
54. Caminero Gomes Soares, A.; Marques Sousa, G. H.; Calil, R. L.; Goulart Trossini, G. H., Absorption matters: A closer look at popular oral bioavailability rules for drug approvals. *Mol Inform* **2023**, *42*, e202300115.
55. Shultz, M. D., Two Decades under the Influence of the Rule of Five and the Changing Properties of Approved Oral Drugs. *J Med Chem* **2019**, *62*, 1701-1714.
56. Stegemann, S.; Moreton, C.; Svanback, S.; Box, K.; Motte, G.; Paudel, A., Trends in oral small-molecule drug discovery and product development based on product launches before and after the Rule of Five. *Drug Discov Today* **2023**, *28*, 103344.
57. Leeson, P. D.; Young, R. J., Molecular Property Design: Does Everyone Get It? *ACS Med Chem Lett* **2015**, *6*, 722-5.
58. Tang, B.; Kramer, S. T.; Fang, M.; Qiu, Y.; Wu, Z.; Xu, D., A self-attention based message passing neural network for predicting molecular lipophilicity and aqueous solubility. *J Cheminform* **2020**, *12*, 15.
59. Wildman, S. A.; Crippen, G. M., Prediction of Physicochemical Parameters by Atomic Contributions. *Journal of Chemical Information and Computer Sciences* **1999**, *39*, 868-873.
60. Roucairol, M.; Cazenave, T. Refutation of Spectral Graph Theory Conjectures with Monte Carlo Search. In *Computing and Combinatorics*, Cham, 2022//, 2022; Zhang, Y.; Miao, D.; Möhring, R., Eds. Springer International Publishing: Cham, 2022; pp 162-176.
61. Roucairol, M.; Cazenave, T. Solving the Hydrophobic-Polar Model with Nested Monte Carlo Search. In *Advances in Computational Collective Intelligence*, Cham, 2023//, 2023; Nguyen, N. T.; Botzheim, J.; Gulyás, L.; Nunez, M.; Treur, J.; Vossen, G.; Koziarkiewicz, A., Eds. Springer Nature Switzerland: Cham, 2023; pp 619-631.

**Table S1:** Time (sec) to generate 1000 molecules.

	random	enforced	ngram	neural
<b>NMCS</b>	1.366	9.395	0.753	3280.072
<b>UCT/PUCT</b>	-	-	0.507	2732.383
<b>sampling</b>	9.154	675.544	0.482	2679.176