Augmented Hill-Climb increases reinforcement learning efficiency for language-based *de novo* molecule generation

Morgan Thomas[1], Noel M. O'Boyle[2], Andreas Bender[1*] and Chris De Graaf[2*]

[1] Centre for Molecular Informatics, Department of Chemistry, University of Cambridge, Cambridge, CB2 1EW, UK

[2] Computational Chemistry, Sosei Heptares, Steinmetz Building, Granta Park, Great Abington, Cambridge, CB21 6DG, UK

E-mail: ab454@cam.ac.uk, chris.degraaf@soseiheptares.com

Keywords: Artificial intelligence; AI; Structure-based drug design; SBDD; Deep learning; Generative models; Recurrent neural network; Molecular docking; Reinforcement learning; *De novo* design; REINFORCE; Hill-Climb; REINVENT

Abstract: A plethora of AI-based techniques now exists to conduct *de novo* molecule generation that can devise molecules conditioned towards a particular endpoint in the context of drug design. One popular approach is using reinforcement learning to update a recurrent neural network or language-based *de novo* molecule generator. However, reinforcement learning can be inefficient, sometimes requiring up to 10^5 molecules to be sampled to optimize more complex objectives, which poses a limitation when using computationally expensive scoring functions like docking or computer-aided synthesis planning models. In this work, we propose a reinforcement learning strategy called Augmented Hill-Climb based on a simple, hypothesis-driven hybrid between REINVENT and Hill-Climb that improves sample-efficiency by addressing the limitations of both currently

used strategies. We compare its ability to optimize several docking tasks with REINVENT and benchmark this strategy against other commonly used reinforcement learning strategies including REINFORCE, REINVENT (version 1 & 2), Hill-Climb and best agent reminder. We find that optimization ability is improved ~1.5-fold and sample-efficiency is improved ~45-fold compared to REINVENT while still delivering appealing chemistry as output. Diversity filters were used, and their parameters were tuned to overcome observed failure modes that take advantage of certain diversity filter configurations. Lastly, we find that Augmented Hill-Climb outperforms the other reinforcement learning strategies used on six tasks, especially in the early stages of training or for more difficult objectives. Overall, we hence show that AHC improves sample-efficiency for language-based *de novo* molecule generation conditioning via reinforcement learning, compared to the current state-of-the-art. This makes more computationally expensive scoring functions, such as docking, more accessible on a relevant timescale.

Introduction

Many generative model techniques and architectures applied to *de novo* molecule generation exist. These models range from purely symbolic approaches such as genetic algorithms [1, 2] to more recent machine learning (ML) approaches such as recurrent neural networks (RNNs) [3–7], transformers [8–10], variational autoencoders [11–14], generative adversarial networks [15–17], graph neural networks [18, 19] and hybrid approaches that use ML to guide reinforcement learning (RL) in a heuristic action space [20]. These generative models can produce valid and novel molecules [21, 22] and condition molecule generation towards a particular endpoint [21] (e.g., predicted bioactivity towards a protein target [4]) via optimization techniques such as, RL [4, 20, 23], Bayesian optimization [11, 13], molecular swarm optimization [14] and Monte Carlo tree search [2, 6]. Although generative models still face many challenges for trusted and routine integration into drug discovery pipelines including practical relevance and more comprehensive evaluation [24].

Of the more recent ML-based approaches to *de novo* molecule generation, RNNs were one of the first to appear with the seminal approaches being published ~ 5 years ago. The first by Segler *et al.* [3] who fine-tuned an RNN on molecules of biological interest to generate molecules containing similar properties *de novo*. The second by Olivecrona *et al.* [4] who instead used RL to update the RNN to generate molecules *de novo* that maximized predicted properties (e.g., predicted bioactivity of molecules). These results were obtained by representing molecules using the SMILES language [25] which emulates the RNN's designed application for use in natural language processing [26, 27]. When trained on a large dataset of SMILES ($>10^5$), an RNN can predict the next symbol in a sequence conditional upon previously seen symbols. Thus, by supplying a start symbol, new symbols can be sampled from the probability distribution corresponding to the next symbol (output by the RNN), which is then recursively fed back into the network resulting in *de novo* molecules. Despite a wave of newer approaches since (e.g., JT-VAE [12], DrugEx [28], GENTRL [29], GraphINVENT [19, 30]), RNNs are still frequently used and investigated for *de novo* molecule generation (e.g., [31–33]). Furthermore, they still match the state-of-the-art on several *de novo* molecule generation benchmarks [21, 22, 34, 35].

Although it is possible to optimize RNN *de novo* molecule generation via fine-tuning on a smaller dataset of molecules relevant to a particular endpoint (as in [3]), such *a priori* knowledge is not always available or may bias *de novo* molecule generation too much towards what is already known resulting in a lack of novelty. Whereas reinforcement learning (RL) can be used to optimize *de novo* compounds to maximize/minimize a numerical reward which can be provided by either a single or a combination of scoring functions. Several RL strategies have been combined with RNNs including Hill-Climb (HC) [21, 36], REINFORCE [37] (used in [5]) and REINVENT [4]. Two of these RL strategies (REINVENT and HC) have been shown to rank top one or two in optimization tasks compared to other generative models [21, 34, 35]. Monte Carlo tree search approaches have also been proposed to search a trained RNN's sample space [6, 38, 39]; however, no RNN parameters are updated

(no RNN learning takes place) during this process and so task optimization is rather an optimized search within the RNNs current generative domain.

Despite excellent performance on benchmarks, RNN *de novo* molecule optimization using RL can be very sample-inefficient often requiring 10s or 100s of thousands of molecules to optimize a task. For example, 163,840 molecules were sampled during HC optimization for GuacaMol benchmark tasks [21] and 192,000 molecules were sampled during REINVENT optimization of DRD2 predicted activity [4] (although neither study specified at which point the task was 'sufficiently' optimized, which could have been before optimization finished). While low sample-efficiency is not a problem for easily computed scoring functions such as property calculation, it significantly hinders the use of scoring functions requiring a significant amount of computation such as molecular docking and computer aided-synthesis planning. This is becoming increasingly important with recent growth in interest in using molecular docking scoring functions to guide *de novo* molecule generation [40–48]. This approach has shown to result in more diverse and novel compounds with a broader coverage of known active space than an equivalent QSAR model trained on known ligands [47]. Other studies have used ML to model molecular docking or other physics-based scoring functions which is less computationally expensive [34, 49, 50]. However, use of a model of a model reduces the advantages of such scoring functions by being less able to extrapolate novel chemical space and adds prediction uncertainty on top of pre-existing inaccuracies [51, 52]. Therefore, it is attractive to improve the sample-efficiency of RL optimization to enable routine use of such docking-based scoring functions directly.

Previous work has explored RL strategies and parameters for RNNs *de novo* molecule generation to varying degrees. Niel *et al.* [36] compared different RL strategies (including REINFORCE, HC and REINVENT) and optimized a selection of tasks. However, the difference in sample-efficiency was not clear and their code was not published. A comparison of REINVENT versions 1.0 and 2.0 shows that the default sigma parameter value was increased. This effectively increases the reward contribution compared to the prior

contribution and theoretically improves sample-efficiency, although this was not discussed in the publication [53]. Fialková *et al.* [54] investigated more significant modifications to the REINVENT loss function which did not result in any significant improvement. Meanwhile, Atance *et al.* [19] modified the loss function by adding a best agent reminder (BAR) mechanism to the loss function resulting in 'significantly improved learning' (although this was not further quantified by the authors and it pertained to use on a graph-based generation model).

Here, with the aim to improve the sample-efficiency of SMILES-based RNNs, we make a very simple change to the REINVENT strategy to ameliorate overpowered regularization by introducing elements of the HC strategy. We call this novel hybrid approach Augmented Hill-Climb (AHC) and investigate it's use for RNN *de novo* molecule generation. We further compare AHC to previously mentioned RL strategies that are implemented in published studies and make the code freely accessible [55].

Methods

The evaluation of AHC and comparison to other RL strategies was built around four key experiments which are summarised in Figure 1 (the details of which follow in the remainder of the Methods): Experiment 1, comparison between AHC and REINVENT on the ability to minimize the docking score against the D_2 receptor (DRD2) over a very limited number RL updates. Experiment 2, comparison between AHC and REINVENT on the ability to minimize the docking score against four different receptors over an extended number of RL updates relative to Experiment 1. Experiment 3, investigation of diversity filters and their parameters for use in combination with AHC by optimizing toy tasks proposed by the GuacaMol benchmark suite [21]. Experiment 4, benchmark comparison between AHC and other RL strategies on six tasks of varying difficulty.

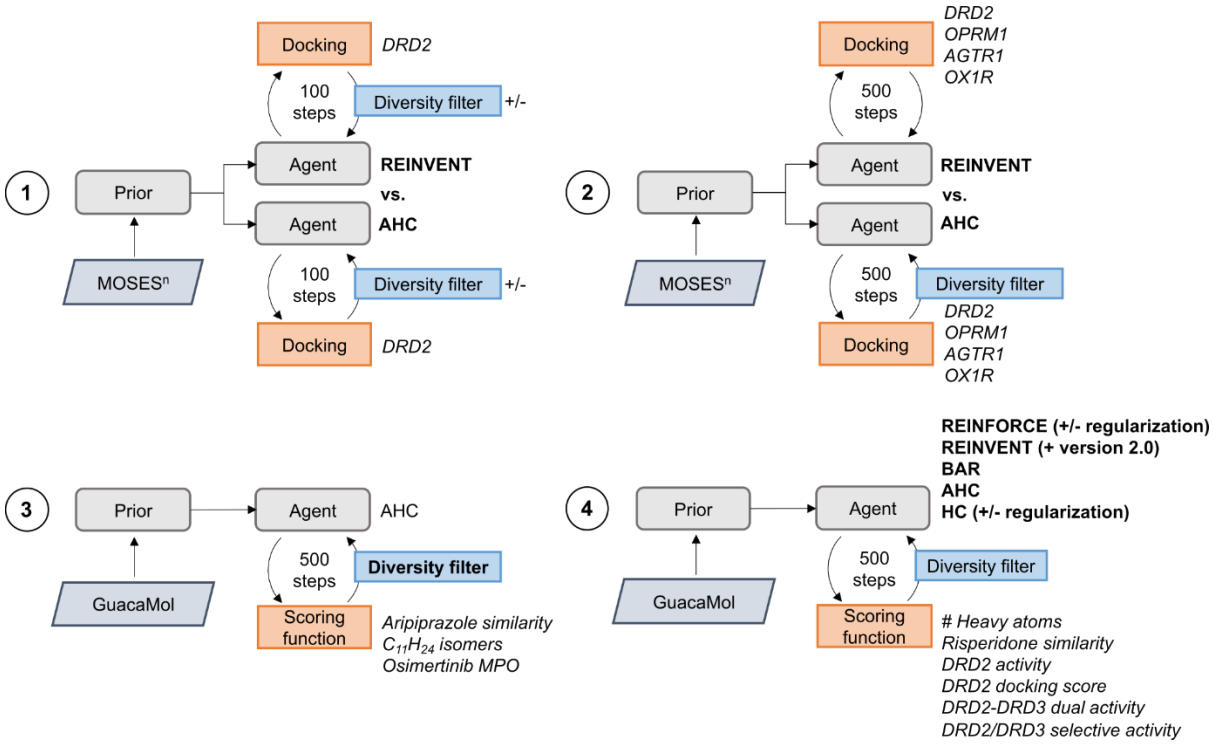

Figure 1: Schematic of the four experiments conducted in this work with the focus of each experiment in bold face. In each case the Prior and Agent refer to an RNN. 1) Comparison of AHC to REINVENT on a single docking task over 100 RL updates. 2) Comparison of AHC to REINVENT on four different docking tasks over 500 RL updates. 3) Diversity filter and parameter search for use in combination with AHC on three toy tasks proposed by GuacaMol benchmark suite. 4) Benchmark comparison of AHC to other RL strategies across a six optimization tasks of varying difficulty.

Recurrent neural network datasets

RNNs were trained using either a modification of the MOSES dataset or the GuacaMol dataset. Firstly, the MOSES dataset [22] is derived from ZINC15 clean leads [56] and contains a library of 'drug-like' small organic molecules. It is designed to benchmark generative model *de novo* molecule generation. The MOSES dataset applies several filters during curation including: molecular weight between 250–350 Da; number of rotatable bonds not greater than 8; XlogP [57] not greater than 3.5; no atoms besides C, N, S, O, F, Cl, Br, H; no cycles larger than 7 members; molecules adhering to custom medicinal chemistry [58, 59] and PAINS filters [60]. In addition, charged species are removed; here however, we deviate from this curation by neutralising charged species and hence avoid a bias towards non-protonatable groups. To distinguish this from the original MOSES dataset, we refer to this as MOSES neutralized ($MOSES^n$)[47]. This resulted in a training set of 2,454,087 molecules. The GuacaMol train dataset [21] (1,273,104 molecules) is derived from ChEMBL24 and contains real molecules both in the 'drug-like' domain and others such as peptides and natural products. This dataset was designed to benchmark both generative model *de novo* molecule generation and subsequent objective optimization. The GuacaMol dataset applies the following filters during curation: salt removal; charge neutralization; molecules with SMILES strings < 100; no atoms besides H, B, C, N, O, F, Si, P, S, Cl, Se, Br, and I. Therefore, the GuacaMol dataset results in a training set with a much broader variety of chemotypes present than $MOSES^n$.

Recurrent neural network

Recurrent neural networks are deep neural networks composed of layers of either long short-term memory units or gated recurrent units, which store and transfer information from one state to the next. In *de novo* molecule generation, SMILES symbols are one-hot encoded into a binary vector which is used as input to the network. These networks are then trained to predict the conditional probability of a SMILES subsequent symbols given a sequence of previously seen SMILES symbols. This is achieved by training the network

using maximum likelihood estimation (equivalent to minimizing the negative log likelihood), whereby the model must maximize the likelihood assigned to the correct symbol $\mathbf{x}$ at time $\mathbf{t}$ conditional upon all previously observed symbols. The resulting loss function $\mathbf{L}$ parameterized by the network parameters $\boldsymbol{\theta}$ is shown in Equation 1. For further details we refer the reader to [4].

$$L(\theta) = -\sum_{t=0}^{T} logP(x_t|x_{t-1} \dots x_0) \quad (1)$$

The RNN implemented in this work is the same as [3, 4, 53, 61]. Specifically, three RNN configurations were used, either trained on MOSES[n] or GuacaMol train. The first RNN configuration consisted of an embedding layer of size 128 and three gated recurrent unit (GRU) layers of size 512 with no dropout – implemented using the code shared in the original work [4]. This implementation was only used with the original REINVENT RL strategy in experiment 2, as a comparison to older work. The second configuration consisted of an embedding layer of size 256 and three long short-term memory (LSTM) layers of size 512 with no dropout – consistent with the REINVENT 2.0 implementation [53]. The third configuration consisted of three LSTM layers of size 512 with a dropout rate of 0.2, consistent with the GuacaMol implementation [21] as found on the corresponding GitHub repository [62]. The first and second configurations were trained on the MOSES[n] dataset for 5 epochs using a batch size of 128 with an ADAM optimizer and learning rate of 0.001, while the third configuration was trained on GuacaMol train for 10 epochs using a batch size of 512 with an ADAM optimizer and learning rate of 0.001.

Reinforcement learning

We will in the following briefly review reinforcement learning strategies for recurrent neural networks in order to embed our methodological changes into context.

RL introduces the paradigm of an episodic task where an agent (here the RNN) decides an action ($\mathbf{a_t} \in \mathbf{A}$) (here, the next SMILES symbol) at time step $\mathbf{t}$ based on interaction with an environment which informs the agent on the current state ($\mathbf{s_t} \in \mathbf{S}$) (here, the SMILES string) and corresponding reward ($\mathbf{r_t}$) (here, computed at the end of the episode ($\mathbf{R_T} \in [\mathbf{0}, \mathbf{1}]$) by the scoring function) in a Markov Decision Process [63]. Different RL strategies can then be used to describe how to navigate this landscape. These usually fall into one of two categories: value-based strategies focus on estimating the value of an action given a particular a state (or value of being in a state) and selecting an action so as to maximize the final estimated return ($G = \sum_t^T r_t$), while policy-based RL focusses on identifying the best policy ($\boldsymbol{\pi}$) for selecting actions without necessarily consulting a value function to estimate the absolute value of that state/action.

The practical nature of SMILES-based RNN molecule generation complicates the use of value-based RL strategies as incomplete SMILES generated at different time steps do not always result in a valid molecule for which a reward can be assigned. In contrast, policy methods do not require a reward for each action/state and as such are typically used in this setting [4, 5, 21]. Furthermore, as discussed by Olivecrona *et al*. [4], an RNN is first trained on a large dataset of example molecules which effectively constitutes a prior policy for molecule generation, thus only small changes to the prior policy may be needed.

As a simple baseline strategy, we implemented REINFORCE [64] which is also used in [36, 65]. This is an 'all-actions' policy-based method because the policy update only requires a sum over all actions and the return for the whole episode (final molecule) – important due to potentially invalid partial smiles during generation. The loss function is described in Equation 2 where it can be interpreted as a scaling of the policy (here the negative log likelihood, also described in Equation 1) by the reward given to the complete molecule ($\mathbf{R_T}$).

$$L(\theta) = \left[-\sum_{t=0}^{T} logP(a_t|s_{t-1}) \right] R_T \quad (2)$$

In this work, we implemented REINVENT [4, 53] (depicted in Figure 2) which is a popular strategy used in the literature, and the strategy we used in our previous work [47]. REINVENT is a REINFORCE type strategy that explicitly regularizes optimization by adding a prior policy to the loss function. This prior policy is derived by computing the negative log likelihood from a fixed copy of the initially trained RNN (the prior). This regularization ensures that the RNN being optimized (the agent) maintains what was initially learnt by the prior i.e., how to generate valid SMILES corresponding to the training distribution. A combination of this prior policy and scaled reward (scaled by scaling coefficient sigma ($\boldsymbol{\sigma}$)) is then used to define an augmented likelihood, as shown in Equation 3. This augmented likelihood then acts as a target policy for the agent and the loss function is now defined as the difference between the agent policy and target policy, shown in Equation 4. Note that we have replaced $-\sum_{t=0}^{T} logP(a_t|s_{t-1})$ by the equivalent term $logP(A)$.

$$logP_{\mathbb{U}}(A) = logP_{prior}(A) + \sigma R_T \quad (3)$$

$$L(\theta) = \left[logP_{\mathbb{U}}(A) - logP_{agent}(A) \right]^2 \quad (4)$$

Recently a strategy was proposed that offered modest performance improvement over REINVENT called ‘best agent reminder’ (BAR) [19], although this was implemented on a graph-based generative model. We have implemented it for an RNN using the same principle to compare it to the other strategies used here as another baseline strategy. This mechanism keeps track of the best agent so far, updating it periodically. During optimization, a batch of molecules $\mathbf{m}$ (of batch size $\mathbf{S}$) is sampled from both the current agent (M_{agent}) and

best agent (M_{best}), to serve as a reminder of high scoring molecules. Although the loss function is the same as Equation 4 for the respective agents, the loss weighted average is taken across agents scaled by **α**, as shown in Equation 5. This effectively acts to minimize the agent policy difference to the 'best agent optimal policy' and the 'prior optimal policy', scaled by α.

$$L(\theta) = \frac{(1-\alpha)}{S} \sum_{m \in M_{agent}} \left[logP_{\mathbb{U}_{prior}}(A) - logP_{agent}(A) \right]^2 + \frac{\alpha}{S} \sum_{m \in M_{best}} \left[logP_{\mathbb{U}_{best}}(A) - logP_{agent}(A) \right]^2 \quad (5)$$

Hill-Climb (HC) [36] is an alternative policy-based strategy benchmarked in [21, 35] that shows state-of-the-art or near state-of-the-art performance. HC can also be interpreted as a form of iterative fine-tuning (where fine-tuning molecules are selected by the scoring function rather than e.g., known activity against a certain target). The agent RNN first samples a batch of molecules, and then the RNN is fine-tuned using the same loss function as Equation 1 but using only the top $\boldsymbol{k}$ molecules from the batch, as ranked according to some reward assigned to each molecule. This algorithm is depicted in the top part of Figure 2.

In this work, we define a new strategy we call Augmented Hill-Climb (AHC), depicted in Figure 2 with its constituent parts shown at the top (for HC), and bottom (for REINVENT). This strategy is a simple hybrid between the HC and REINVENT strategies where the loss is calculated as in REINVENT (by defining the augmented likelihood) but only on the top $\boldsymbol{k}$ molecules, as ranked by reward as in HC. The rationale behind this strategy is based on practical limitations of the REINVENT loss function: when low scoring molecules ($R_T \to 0$) are sampled the score contribution goes to zero and $logP_{\mathbb{U}}(A) \approx logP_{prior}(A)$. In this situation, as the loss function (Equation 4) is a distance, the agent policy will, in-fact, trend back towards the prior policy which may negate useful learnings. This situation of low scoring molecules being present will occur especially either early in the learning process or

when a difficult or highly constrained scoring function is used. Therefore, the heavy regularization effect of low scoring molecules significantly contributes to slow learning in these situations. In turn, focussing learning only on the high scoring molecules ($R_T \rightarrow 1$) will improve learning. It is worth noting that, high scoring molecules are still regularized by the prior policy, as shown in Equation 3, ensuring prior learnings are not 'forgotten'.

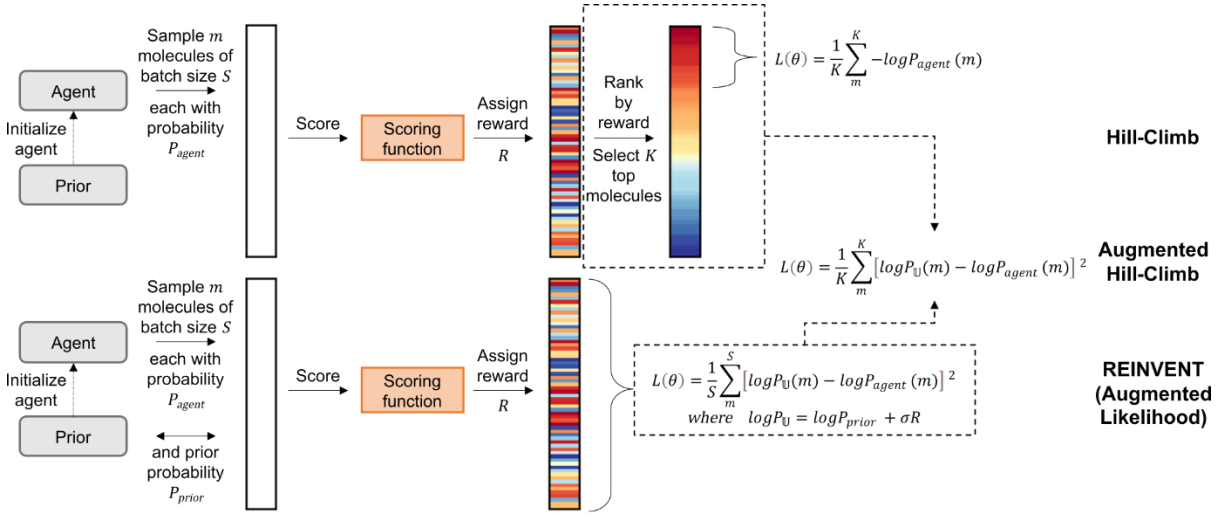

Figure 2: Depiction of the REINVENT, Hill-Climb (HC) and Augmented Hill-Climb (AHC) optimization algorithms and subsequent loss functions $\boldsymbol{L}$ as parameterized by network parameters $\boldsymbol{\theta}$. AHC is a hybrid algorithm that combines elements of REINVENT and HC.

As the RL strategies REINFORCE and HC are not explicitly regularized (as they are in REINVENT, BAR and AHC), cost terms can be added to the loss function to achieve regularization. This step is important in practice to maintain some similarities to the training distribution but also to not catastrophically forget chemical principles which will result in invalid structures (due to valency errors etc.). To assess the effectiveness of this, we evaluated the addition of the Kullback-Leibler (KL) divergence between the prior and agent scaled by a scaling coefficient $\boldsymbol{\lambda}$, as shown in Equation 6 and as implemented in [36, 66]. This adds a constraint to ensure the distribution of agent action probabilities does not differ too much from the distribution of prior action probabilities.

$$C(KL) = \lambda_{KL}\mathbb{E}\left[\sum_{t=0}^{T}\sum_{a_i \in A} P_{agent}(a_i|s_{t-1}) log \frac{P_{agent}(a_i|s_{t-1})}{P_{prior}(a_i|s_{t-1})}\right] \quad (6)$$

Unless otherwise specified, the hyperparameters used for the different RL strategies are those reported in each individual study. They are listed in Table S1. The number of RL update steps was adjusted to result in an approximately equal number of molecules sampled during training. Hill-Climb* was included to investigate the effect of a smaller batch size in line with AHC.

Diversity filters

Applying diversity filters (DFs) is a way of penalizing the reward for an associated molecule based on the molecular similarity to previously generated molecules resulting in diminishing returns for exploitation, therefore encouraging exploration outside of local minima. Blaschke *et al.* [67] introduced several DFs for RNN molecule generation based on different measures of similarity including *compoundsimilarity* (Tanimoto similarity of compound ECFP [68]fingerprints), *identicalmurckoscaffold* (matching Bemis-Murcko scaffolds [69]), *identicaltopologicalscaffold* (matching Bemis-Murcko scaffolds with all atoms treated as carbon atoms and bonds as single bonds) and *scaffoldsimilarityatompair* (Tanimoto similarity of scaffold atom pair fingerprints [70]). More specifically, if generated molecules receive a high enough score by a scoring function (*minimum score threshold*) then the molecules are added to bins based on similarity as defined by any of the above-mentioned DFs. Molecules assigned to a bin are subsequently penalized according by a binary, sigmoid or linear score transformation (*output mode*) based on the maximum allowed *bin size*. Blaschke *et al.* [67] showed that they result in increased diversity of *de novo* compounds as measured by an increased number of analogues to known molecules.

In addition, we investigated the use of the following DFs:

1) *unique* – a simple DF to serve as a baseline. This DF transforms a molecule's score to zero if the molecule is non-unique.

2) *occurrence* – This DF linearly penalizes non-unique molecules based on the number of previous occurrences, which acts as a more lenient version of the *unique* DF. The score is transformed according to the number of previous occurrences (**Occ**) beyond an allowed tolerance (**Tol**) until a hard threshold is reached, referred to as the buffer (**Buff**). This is shown in Equation 7.

$$\text{Filtered reward} = \begin{cases} R_T \times \dfrac{\text{Occ-(Tol+Buff)}}{\text{Tol+Buff}} & \text{if } \text{Tol} < \text{Occ} < \text{Buff} \\ R_T & \text{if } \text{Occ} \leq \text{Tol} \\ 0 & \text{if } \text{Occ} \geq \text{Buff} \end{cases} \quad (7)$$

3) *scaffoldsimilarityecfp* – This DF is a modification to those *scaffoldsimilarityatompair* introduced in [67] that uses the same parameters except for measuring similarity based on the Tanimoto similarity of the Bemis-Murcko [69] scaffold ECFP4 [68] fingerprints as implemented by RDKit [71].

The DFs and parameters used in this work (i.e., DF1, DF2 and DF3) for tasks other than the parameter search in Experiment 3 are shown in Table S2.

Scoring functions and benchmarking tasks

Several scoring functions were used in this work to guide optimization and benchmark RL strategies. These are summarized in Table 1 and are described in more detail in the subsequent sections. All scoring functions were implemented using the MolScore platform [55] (manuscript in preparation).

Table 1: Summary of all objectives / tasks used in this work and for which experiment (see Figure 1).

Experiment	Aim	Objective type	Objective target	Performance measure
1	Compare REINVENT and AHC for varying values of σ	Docking	DRD2	Docking score & uniqueness
2	Compare REINVENT and AHC against different target systems	Docking	DRD2	Docking score & uniqueness
		Docking	OPRM1	Docking score & uniqueness
		Docking	AGTR1	Docking score & uniqueness
		Docking	OX1R	Docking score & uniqueness
3	Investigate and identify optimal DF and respective parameters for use with AHC	Similarity	Aripiprazole	Tanimoto similarity, uniqueness & wall time
		Isomer	$C_{11}H_{24}$	Isomer score, uniqueness & wall time
		Similarity & PhysChem (MPO)	Osimertinib	MPO score, uniqueness & wall time
4	Benchmark AHC to other commonly used RL strategies	PhysChem	Heavy atoms	# Heavy atoms, validity, uniqueness & wall time
		Similarity	Risperidone	Tanimoto similarity, validity, uniqueness & wall time
		Activity	DRD2	Predicted activity, validity, uniqueness & wall time
		Docking	DRD2	Docking score, validity, uniqueness & wall time
		Dual activity (MPO)	DRD2 & DRD3	Average predicted activity, validity, uniqueness & wall time
		Selectivity (MPO)	DRD2 > DRD3	Average predicted activity, validity, uniqueness & wall time

Target preparation and docking tasks

Four different targets were used to setup molecular docking scoring functions to evaluate docking score optimization by RNNs in combination with RL strategies (Experiments 1, 2 and 4 in Figure 1). The four targets and corresponding x-ray crystal structures used in the docking tasks were D_2 (DRD2, PDB: 6CM4 [72]), μ (OPRM1, PDB: 4DKL [73]), AT_1 (AGTR1, PDB: 4YAY [74]) and OX_1 (OX1R, PDB: 6TO7 [75]) receptors.

All target crystal structures were first prepared using Schrodinger Protein Preparation Wizard [76] using default parameters which included: addition of protein and ligand hydrogens (pH 7±2, Epik [77]), optimization of hydrogen bond networks (pH 7, PROPKA [78]), restrained minimization using the OPLS3e force field [79], and waters except for OPRM1 (which performed better retrospectively with crystallographic waters, data not shown). A default grid was defined using the respective co-crystallized ligands as the centre except for OX1R which had additional positional restraints defined based on consensus sub-pocket occupation by the following overlayed co-crystallized ligands, Suvorexant (PDB: 6TO7), Filorexant (PDB: 6TP6), Daridorexant (PDB: 6TP3), GSK1059865 (PDB: 6TOS), ACT462206 (PDB: 6TP4), Compound-16 (PDB: 6TQ4), Compound-14 (PDB: 6TQ6), EMPA (PDB: 6TOD) and Lemborexant (PDB: 6TOT) [75].

Before docking, ligands were prepared using Schrodinger LigPrep [80] to enumerate unspecified stereocentres, tautomers and protonation states, with up to 8 variants generated per molecule, based on a pH range of 7±1. Variants were then docked using Glide-SP [81] with default settings, except for OX1R where docked poses were only accepted if they satisfied four out of five grid constraints. The lowest (i.e., best) docking score achieved by any molecule variant was returned as the final docking score. Docking score was normalized between the values of 0 and 1 based on all previously observed docking scores.

Retrospective performance was assessed by docking known active and inactive molecules extracted for each human target from the ExCAPE-DB [82]. When more than 10,000 labelled molecules were present, a random subset of 10,000 molecules was taken. To better

represent *de novo* molecules docked which adhere to property constraints imposed by MOSES[n], molecules above 500 Da were filtered out, stereo information removed, and any resulting duplicates removed. The final number of downloaded and docked molecules is shown in Table S3. Classification accuracy, precision and recall were assessed by varying docking score decision thresholds (Figure S1). In each case a threshold corresponding to ~80% precision was identified, i.e. ~80% of molecules below this threshold are true actives retrospectively. The typical recall of true actives at this level was ~10-30%.

Diversity filter parameter optimization tasks

To investigate the effect of DF and parameter choice, less computationally expensive scoring functions were required than docking. Therefore, three diverse tasks from the GuacaMol benchmarking suite [21] were chosen and re-implemented according to the original work [21]. The goal the Aripiprazole similarity task is to optimize similarity to Aripiprazole beyond a similarity threshold in order to generate as many similar enough compounds as possible. The goal of the $C_{11}H_{24}$ isomer task is to generate all 159 molecules with a molecular formula of $C_{11}H_{24}$, a task involving a more limited pool of molecules. The goal of the Osimertinib MPO task is to optimize similarity to Osimertinib to a certain extent, while ensuring molecules are not too similar and that both lipophilicity and polarity are within a suitable range. The performance of DF parameters was measured by the area under the training curve of three different endpoints: uniqueness (number of unique molecules generated, a proxy of chemical space explored), goal (the score returned by the scoring function/s) and run time (a practical measure to identify if some DFs are slower to compute).

QSAR model training

Active and inactive molecules against DRD2 and against DRD3 were extracted from the ExCAPE-DB [82]. This corresponded to 4,609 and 2,758 active molecules and 343,026 and 402,524 inactive molecules respectively. A further unique subset was defined for each target by excluding molecules with measured activity against the other target to ensure no domain overlap between DRD2 and DRD3 models for the dual and selective tasks, resulting in in

2,282 and 373 active molecules and 5,161 and 64,717 inactive molecules for DRD2 and DRD3 respectively. To tackle data imbalance, a maximally diverse selection of 5,000 inactive molecules were selected for DRD2 and DRD3, respectively, *via* a MaxMin algorithm [83] on ECFP4 fingerprints with 2,048 bits, implemented in RDKit. Three random forest (RF) classification models were trained to predict probability of activity (with 100 estimators, max depth of 15 and minimum leaf sampled of 2), one on all DRD2 data with the diverse inactive subset and two on DRD2 and DRD3 unique data with diverse inactive subsets, all implemented in scikit-learn [84]. In each case model performance was estimated by stratified, active cluster split (inactive molecules were split randomly due to being a maximally diverse selection) 5-fold cross-validation with GHOST decision threshold identification [85] resulting in the performance shown in Figure S2.

Reinforcement learning strategy benchmark tasks

Six further tasks of varying difficulty were used to benchmark the different RL strategies at three levels of objective complexity:

1) *# Heavy atoms* – This 'easy' task aims to maximize the number of heavy atoms in a molecule calculated by RDKit [71]. This probes the RL strategy's ability to extrapolate beyond the training dataset which contains molecules with a limited number of heavy atoms. However, this task is irrelevant to real drug discovery objectives.

2) *Risperidone similarity* – This 'easy' task aims to maximize the Tanimoto similarity to Risperidone (a DRD2 inverse agonist and co-crystallized ligand in PDB: 6CM4) according to ECFP4 fingerprints with a bit length of 1,024 (as implemented in RDKit). While this tests the ability to move to a precise region of chemical space, it is unlikely to be relevant as a real drug discovery objective due to lack of novelty.

3) *DRD2 activity* – This 'medium' task aims to maximize the QSAR predicted probability of activity against DRD2 (Equation 8). This task is representative of a real objective during early-stage hit finding, providing that known ligand data is available.

$$DRD2\ active = P_{RF}(DRD2) \quad (8)$$

4) *DRD2 docking score* – This 'medium' task aims to minimize the Glide-SP docking score (predicted binding affinity) against DRD2. This task is representative of a real objective during early-stage hit finding, providing that a crystal structure or homology model is available. It was implemented as described above with the exception that molecules were instead prepared by enumerating up to 16 stereoisomers using RDKit [71] and then conducting protonation using Epik (pH 7.4) to only protonate the most abundant state per stereoisomer.

5) *DRD2-DRD3 dual* – This 'hard' task aims to maximize the QSAR predicted probability of activity against both DRD2 and DRD3 (Equation 9). This task is representative of real drug discovery projects requiring polypharmacological activity, providing that ligand data for both is available.

$$DRD2 - DRD3\ dual = \frac{P_{RF}\left(DRD2_{unique}\right) + P_{RF}(DRD3_{unique})}{2} \quad (9)$$

6) *DRD2/DRD3 selective* – This 'hard' task aims to maximize the QSAR predicted probability of selective activity against DRD2 over DRD3 (Equation 10). This is representative of real drug discovery projects that must avoid off-target effects for toxicity or efficacy reasons, providing that ligand data for both is available.

$$DRD2/DRD3\ selective = \frac{P_{RF}(DRD2_{unique}) + (1 - P_{RF}(DRD3_{unique})}{2} \quad (10)$$

Results & Discussion

Optimization of DRD2 docking score by Augmented Hill-Climb compared to REINVENT

Optimization ability and sample-efficiency was assessed using the procedure described in Methods (Experiment 1, Figure 1). Specifically a RNN was trained on the MOSES^n dataset [22, 47], an agent was initialized which then underwent RL updates to optimize the docking score of de novo molecules against DRD2. The REINVENT strategy and docking protocol was identical to our previous work [47].

To increase optimization power, the easiest proposal is to increase the score contribution to the augmented likelihood used by REINVENT by increasing the scalar value σ. The original work [4] had a default value of 60, however, the subsequent update (REINVENT 2.0 [53]) increased this value to 120 - suggesting that sample-efficiency was sub-optimal. Therefore, we first varied the value of σ between 30 and 240 and updated an agent for 100 RL steps only (6,400 samples), to minimize computational expense. However, as shown in Figure 3a, we found little improvement in optimization of DRD2 docking scores using this approach with REINVENT. The maximum docking score optimization achieved (best mean score relative prior mean score) was 128% with σ=60 or 127% with σ=240, concluding that changing σ values alone did not significantly improve optimization over limited RL updates.

AHC was then implemented in an effort to improve sample-efficiency, while also varying σ for over the same amount of RL updates (Figure 3a). This consistently led to improved optimization ability for every σ value compared to REINVENT, with a maximum of 205% optimization with σ=240. In total, we found a 1.39-fold improvement in optimization ability compared to REINVENT averaged across all values of σ. Moreover, AHC required approximately 80 fewer steps to achieve the mean docking score achieved by REINVENT over 100 steps, evidencing a large improvement in sample-efficiency. However, learning was

stifled by a drop in uniqueness observed (Figure 3b) i.e., AHC was more prone to mode collapse.

To address the mode collapse, a diversity filter (DF1) [67] was applied to both strategies to penalize exploitation and hence encourage exploration. DF1 penalizes the score of any of the top 20% of de novo molecules that were similar to previously generated molecules, a threshold chosen based on the nature of docking-based virtual screening where only the very top ranked molecules are considered. This stabilized learning and rescued the drop in uniqueness in most cases (Figures 3c and 3d). With DF1, AHC evidenced a σ-averaged 1.45-fold improvement compared to REINVENT (with a maximum optimization of 192% at σ=180 for AHC, compared to 119% at σ=180 for REINVENT). Similar to without the DF1, AHC still required 80-90 fewer RL steps to achieve a mean docking score achieved by REINVENT over 100 steps.

Although increasing the σ value increases the score contribution to the loss, it also decreases the prior contribution and thus decreases regularization during optimization. As such, we expect that larger values of σ result in further extrapolation outside the domain of the training set and prior, which is the aspect of the generated molecules we analysed next. Figures 3e-g show the properties of de novo molecules generated during optimization and the property space not occupied by molecules in the MOSES^n dataset – serving as a proxy to assess extrapolation. AHC in combination with DF1 is more sensitive to changes in σ, where larger values of σ do result in extrapolation into property space that is absent in MOSES^n, more so than REINVENT in combination with DF1. In practice, this extrapolation can be both favourable (by identifying novel chemical space) or unfavourable (by enabling exploitation of scoring function flaws, such as molecules with more heavy atoms providing better docking scores simply due to the additive nature of docking scoring functions [86]). In either case, it is advantageous to have greater control over this trade-off, which is achieved as variations in σ show more impact for AHC over REINVENT. Importantly, AHC still

improves 1.47-fold over REINVENT at σ=60, where both strategies are sufficiently regularized and maintain the property space as defined by MOSES[n].

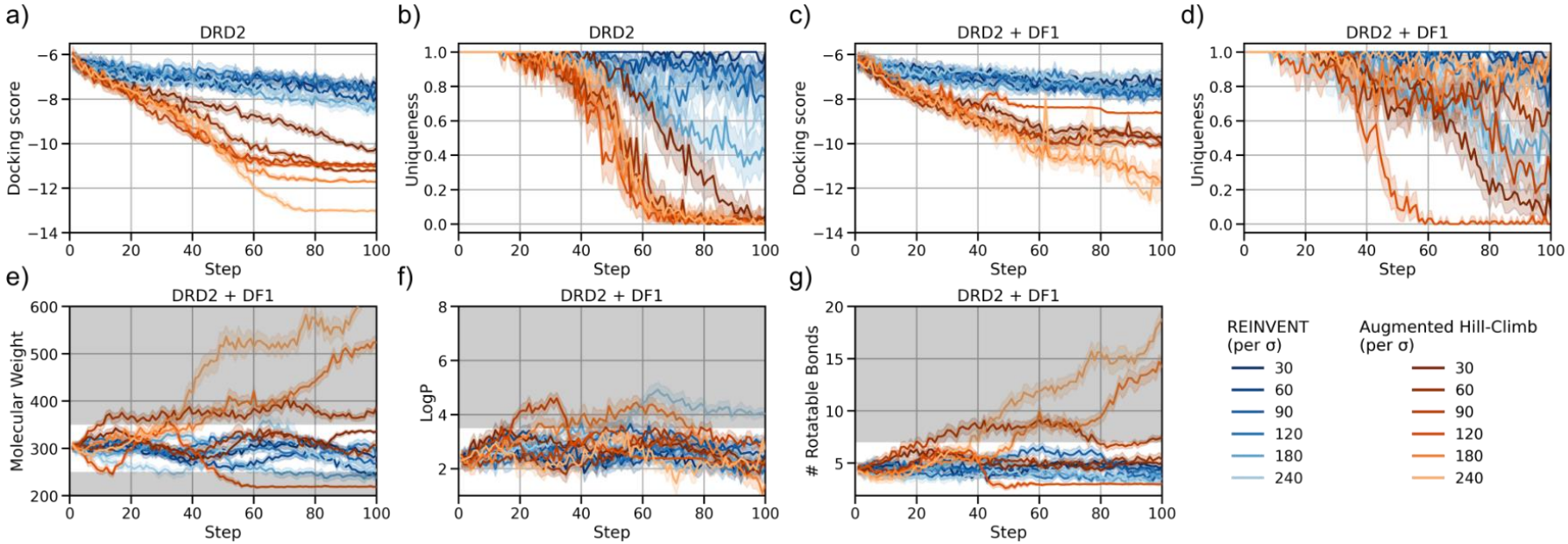

Figure 3: Comparison between REINVENT and Augmented Hill-Climb learning strategies to optimize DRD2 docking scores at varying levels of σ. (**a**) Augmented Hill-Climb is more efficient at optimizing docking score at all levels of σ but (**b**) undergoes increased mode collapse via a drop in uniqueness. (**c**) Docking score optimization can be stabilized and (**d**) mode collapse rescued by applying a diversity filter. (**e-g**) Augmented Hill-Climb in combination with DF1 is more sensitive to changes in σ, this affects the extent to which de novo molecules occupy property space which is not present in the prior training set (grey shaded area) i.e., extrapolation.

Despite improvement in the optimization ability by AHC, it is irrelevant if the resulting de novo structures are invalid or implausible (e.g., incorrect valences, unstable or idiosyncratic functional groups or strained ring systems). The chemistry generated by RNNs has been evaluated previously [3, 22, 32, 87, 88] and has usually been considered reasonable with respect to overall topology, fragments, substructures and property space. On the other hand, a comparison of chemistry between AHC and REINVENT is complicated by the scoring function and its suitability for an objective e.g., greater optimization may actually lead to unreasonable chemistry due to scoring function exploitation rather than as a function of the RL strategy. We note that this analysis of scoring function suitability is out of the scope of this work but we aim to cover this in future work. On the other hand, the REINVENT strategy has been shown to maintain similar chemistry to the prior RNN [4, 47, 48, 67]. Therefore, we visually compared some of the top molecules generated at different values of σ, shown in Figure S3. At lower values of σ (30-120) and with no regard for prior knowledge of DRD2 ligand topology, the molecules are mostly indistinguishable as to which RL strategy was used. With regard for DRD2, both strategies learn to generate benzyl / bicyclic moieties with a protonatable amine above. This chemotype is consistent with the co-crystallised inverse agonist risperidone [72] and required interactions to D114^{3x32} for ligand activity [89–91], where the cyclic moiety would sit deep in the hydrophobic sub-pocket and the amine would form a salt bridge with D114^{3x32}. The only difference between the RL strategies appears to be the better docking scores achieved by AHC. However, as σ increases (180-240), de novo molecules are clearly much larger and therefore exploiting the additive nature of the docking scoring function [86]. This corroborates the observation of extrapolation into restricted property space seen in Figure 3e and g, which enables this exploitation. In this scenario added constraints would be necessary in a multi-parameter optimization setting, such as also defining a suitable molecular weight range as this knowledge is no longer imposed by the prior dataset. We believe these results highlight the balance that is required in the trade-off between regularization and optimization, which is better achieved by AHC than REINVENT.

Optimization of docking scores for multiple GPCR targets

Previously, we used REINVENT to optimize the docking score against other GPCR targets (DRD2, OPRM1, AGTR1 and OX1R) over the course of 3,000 RL updates, the first 500 updates of which are shown in Figure 4. DRD2 [72] (same data as previously published [47]) contains a deep hydrophobic sub-pocket and requires a salt bridge interaction with $D114^{3x32}$ for ligand activity. OPRM1 [73] similarly forms a salt bridge interaction via $D147^{3x32}$ (a structurally conserved position in aminergic receptors [89, 90]) but with a more open pocket than DRD2. AGTR1 [74] requires important salt bridge and hydrogen bond interactions to $R167^{4x65}$ (e.g., via acidic tetrazole of co-crystallised ligand ZD7155) as well as hydrogen bonds to $Y35^{1x39}$ on the opposite side of the pocket. Meanwhile OX1R [75] contains four well defined hydrophobic sub-pockets and sometimes a hydrogen bond to $N318^{6x55}$ and water mediated hydrogen bond to $H344^{7x38}$, ligands are found to adopt a horseshoe conformation via π-stacking to satisfy these sub-pockets as in the co-crystallised ligand suvorexant. The first two targets' respective docking scores were able to be minimized similarly (Figure 4a and 4b), while the latter two targets' respective docking scores were more challenging and showed little minimization (Figure 4c and 4d) (especially with respect to the distribution of docking scores for known actives). This suggests that the docking score optimization ability of REINVENT was system dependent or that the MOSES[n] dataset used for RNN pretraining did not contain chemistry amenable to minimizing the docking score for these systems.

Given the improved optimization power of AHC in combination with DF1 seen with fewer RL updates against DRD2, AHC in combination with DF1 was compared to these REINVENT results to see if improvement was consistent over 500 RL updates and for different GPCR targets (Experiment 2, Figure 1). For every target, AHC in combination with DF1 (Figure 4) resulted in faster and further minimization of the docking score. For reference, the 80% retrospective precision threshold was surpassed within 100 RL updates in all cases except for the particularly challenging OX1R. However, docking score plateaus for AHC in combination with DF1 in later stages of training. This plateau signals mode collapse as

uniqueness drops, similar to training without a DF as shown in Figure 3a. Interestingly, a convergence of the normalized docking score towards the minimum score threshold of the DF occurs, and uniqueness then drops for all targets (Figure S4a). It appears that the model learns to generate molecules with a score just below the minimum score threshold to avoid DF penalization and is thus vulnerable to mode collapse as observed without the DF (Figure S2a-b).

Therefore, we conducted a search of DFs and parameters to identify a more optimal configuration that would successfully and robustly rescue mode collapse (Experiment 3 in Figure 1). Various DF parameters were tested against 3 example optimization objectives taken from the GuacaMol benchmark suite [21]. The prior was therefore trained on the GuacaMol train set with and identical RNN configuration to the GuacaMol LSTM baseline model and trained for the same number of epochs [62]. The tasks were optimized using AHC in combination with DF1 for 500 RL updates. The area under the training curve of three endpoints measured (uniqueness, goal/score and run time) are shown in Figures S5-7. In all cases, we found that too high a minimum score threshold (> 0.5) leads to poorer performance. For uniqueness, linear and sigmoid output modes performed best (and better with lower bin sizes) in the Aripiprazole and Osimertinib tasks. However, with respect to the objective, there was less discrepancy between output modes and the bin size relationship reversed (with higher bin sizes showing better performance). In these tasks, the *identicalmurckoscaffold* and *scaffoldsimilarityecfp* DFs outperformed the other DFs, while *scaffoldsimilarityatompair* seemed to result in an unusually long run time. Based on these results, as well as the rationale of softening the gradient of penalization, we decided to continue using *scaffoldsimilarityecfp* but lowered the minimum score threshold to 0.5, changed the output mode to linear and increased the bin size to 50. This configuration is from here on referred to as DF2.

Using DF2 we re-ran the previous experiment on the four targets as before, shown in Figure 4. The change in DF stabilized learning over the full length of training while still resulting in

similar optimization of docking score. Moreover, there was no convergence of normalized docking score to the minimum score threshold and thus uniqueness stayed relatively high (Figure S4b). To gain a quantitative understanding of improvement in sample-efficiency, Table 2 compares the number of steps (and samples) required by AHC in combination with DF2 and REINVENT to reach various thresholds during optimization. This shows that the largest improvement over REINVENT is made early, where AHC in combination with DF2 requires 19.8-fold fewer training steps until the mean surpasses 120% optimization, however, both strategies sample a single molecule with a docking score exceeding this threshold within the first batch. Meanwhile, AHC in combination with DF2 took 71.8-fold fewer samples than REINVENT until a molecule surpassed 160% optimization. At 180% and 200% optimization, REINVENT only sampled molecules surpassing the threshold for OX1R and thus fold-improvement could not be calculated, however a minimum estimate is shown based on the maximum number of training steps or samples generated. On average, AHC in combination with DF2 required 7.4-fold fewer training steps and 45.5-fold fewer samples across all targets and all optimization thresholds.

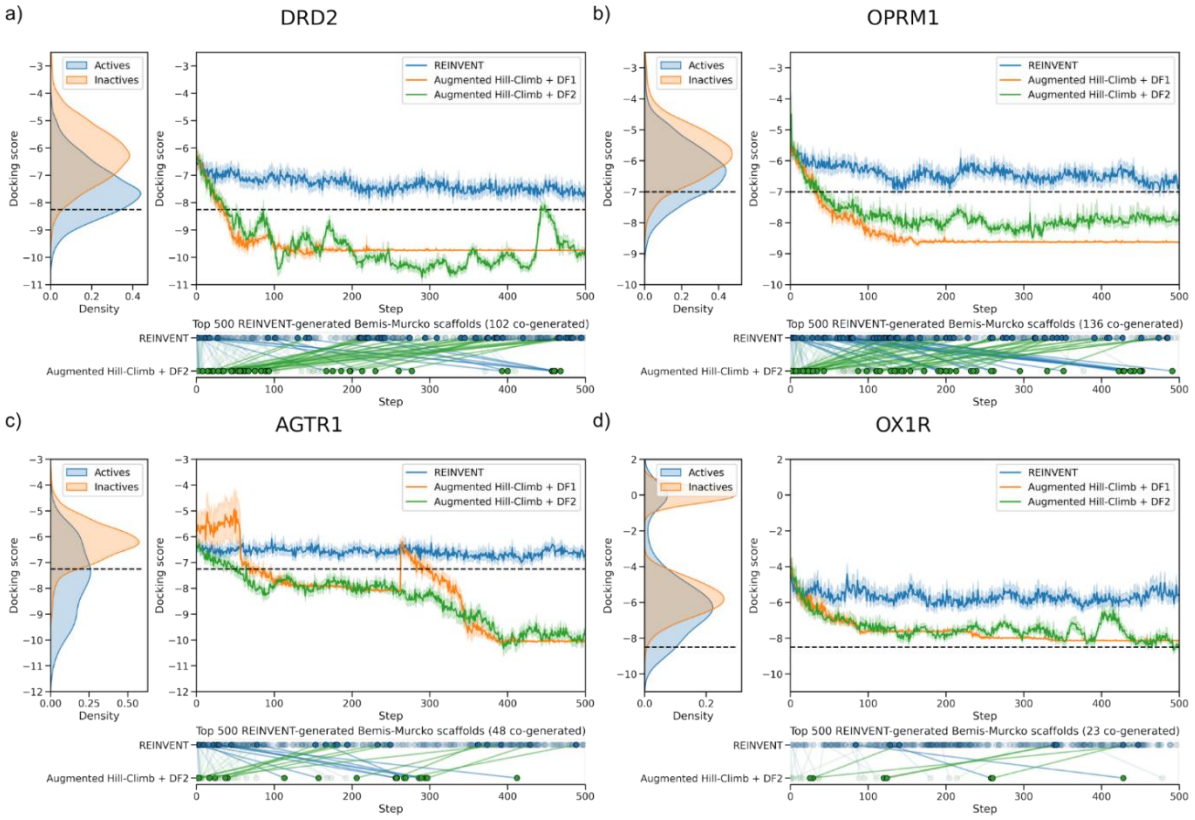

Figure 4: Improved learning efficiency of Augmented Hill-Climb against four targets: (**a**) DRD2, (**b**) OPRM1, (**c**) AGTR1 and (**d**) OX1R. (top left panel) Distribution of known active and inactive molecule docking scores. (top right panel) Optimization of de novo molecule docking score via reinforcement learning. (bottom right panel) The top 500 REINVENT generated scaffolds with the corresponding time of generation by REINVENT or by Augmented Hill-Climb (in combination with DF2) if co-generated. Blue lines represent scaffolds generated by REINVENT first and green lines generated by Augmented Hill-Climb (in combination with DF2) first. Scaffolds with a difference in generation time of < 100 RL updates are more transparent. Augmented Hill-Climb in combination with DF2 shows improved learning efficiency compared to REINVENT and optimizes past a docking score threshold corresponding to a retrospective classification precision of 80% (black dashed line) in all cases.

589 Table 2: Number of steps taken before the mean exceeds certain internal and external thresholds (earliest sample exceeding threshold is shown in brackets).
590 The final row lists the Augmented Hill-Climb in combination with DF2 fold improvement over REINVENT. Where a threshold was not reached within the
591 maximum number of training steps (or samples) it has been annotated as being greater than 500 (or 32,000).

592

		Number of steps required for optimization beyond prior at a given threshold					Number of steps required for optimization beyond external thresholds		
	Threshold	120%	140%	160%	180%	200%	Inactive mean	Active mean	80% precision threshold
DRD2	REINVENT	> 500 (15)	> 500 (685)	> 500 (22,292)	> 500 (> 32,000)	> 500 (> 32,000)	1 (1)	163 (15)	> 500 (15)
	Augmented Hill-Climb + DF2	19 (2)	6 (49)	105 (1,248)	> 500 (3,009)	> 500 (23,150)	2 (2)	19 (2)	48 (2)
OPRM1	REINVENT	133 (7)	> 500 (868)	> 500 (7,663)	> 500 (> 32,000)	> 500 (> 32,000)	4 (2)	80 (4)	> 500 (7)
	Augmented Hill-Climb + DF2	3 (16)	17 (22)	45 (29)	150 (34)	> 500 (2,759)	6 (16)	17 (22)	33 (28)
AGTR1	REINVENT	> 500 (25)	> 500 (510)	> 500 (5,596)	> 500 (> 32,000)	> 500 (> 32,000)	1 (2)	> 500 (8)	419 (6)
	Augmented Hill-Climb + DF2	62 (27)	318 (869)	396 (3,404)	> 500 (5,207)	> 500 (27,979)	2 (1)	62 (27)	46 (2)
OX1R	REINVENT	5 (1)	52 (1)	> 500 (7)	> 500 (142)	> 500 (490)	1 (2)	9 (1)	> 500 (490)
	Augmented Hill-Climb + DF2	9 (1)	15 (2)	31 (2)	87 (31)	382 (557)	2 (1)	14 (2)	494 (557)
	Average fold improvement	19.8 (2.5)	11.2 (38.7)	8.3 (71.8)	2.8 (240.6)	1.1 (3.8)	0.5 (1.0)	5.5 (2.1)	9.7 (3.2)

To investigate if similar chemistry was generated by the RL strategies, we identified the top 500 scaffolds generated by REINVENT for each target and plotted at what stage they were first generated by either RL strategy, shown in Figure 4 (bottom part of each panel of the figure). This shows a general trend where AHC in combination with DF2 tends to generate scaffolds appearing in REINVENT at a later stage much sooner, and scaffolds appearing in REINVENT much earlier. That is, AHC in combination with DF2 identifies chemistry where the mean docking score has improved more than 100 steps sooner, while early chemistry typically achieved due to batch variance more than 100 steps later – likely because of the DF encouraging exploration and re-visiting sub-optimal chemistry.

A visual comparison of the centroids of the top 100 compounds for each target for AHC in combination with DF2 and REINVENT is shown in Figure 5. With disregard to prior knowledge of target ligands and suitability of the scoring function, the quality of chemistry generated is again indistinguishable between the two RL strategies. However, regarding co-crystal ligands and known important residue interactions, the scoring function is not always suitable as shown in the case of AGTR1. Here we can see no acid moieties are generated for AGTR1 by either strategy (Figure 5) which will be in part due to the docking algorithm targeting only the $Y35^{1x39}$ sub-pocket and out towards the extracellular surface (Figure S8c) as opposed to the sub-pocket surrounding $R167^{4x65}$ as required for ligand activity [74].

In addition, we investigated property space occupied by AHC generated *de novo* molecules (Figure 6) which shows that the property space is still maintained in all cases except for increasing molecular weight seen with OX1R. Here, the mean is slightly above 350 Da which is however consistent with OX1R antagonists [75]. In fact, in some cases (for OPRM1 in the case of molecular weight and number of rotatable bonds, and for OX1R in the case of the number of rotatable bonds) the property space shifts in the opposite direction to that which would be expected by an exploitation of the scoring function. Overall, de novo chemistry is still reasonable and sufficiently regularized by AHC in combination with DF2 and can even

be more heavily regularized by reducing σ to 30, yet still outperform REINVENT at all σ values as seen in Experiment 1.

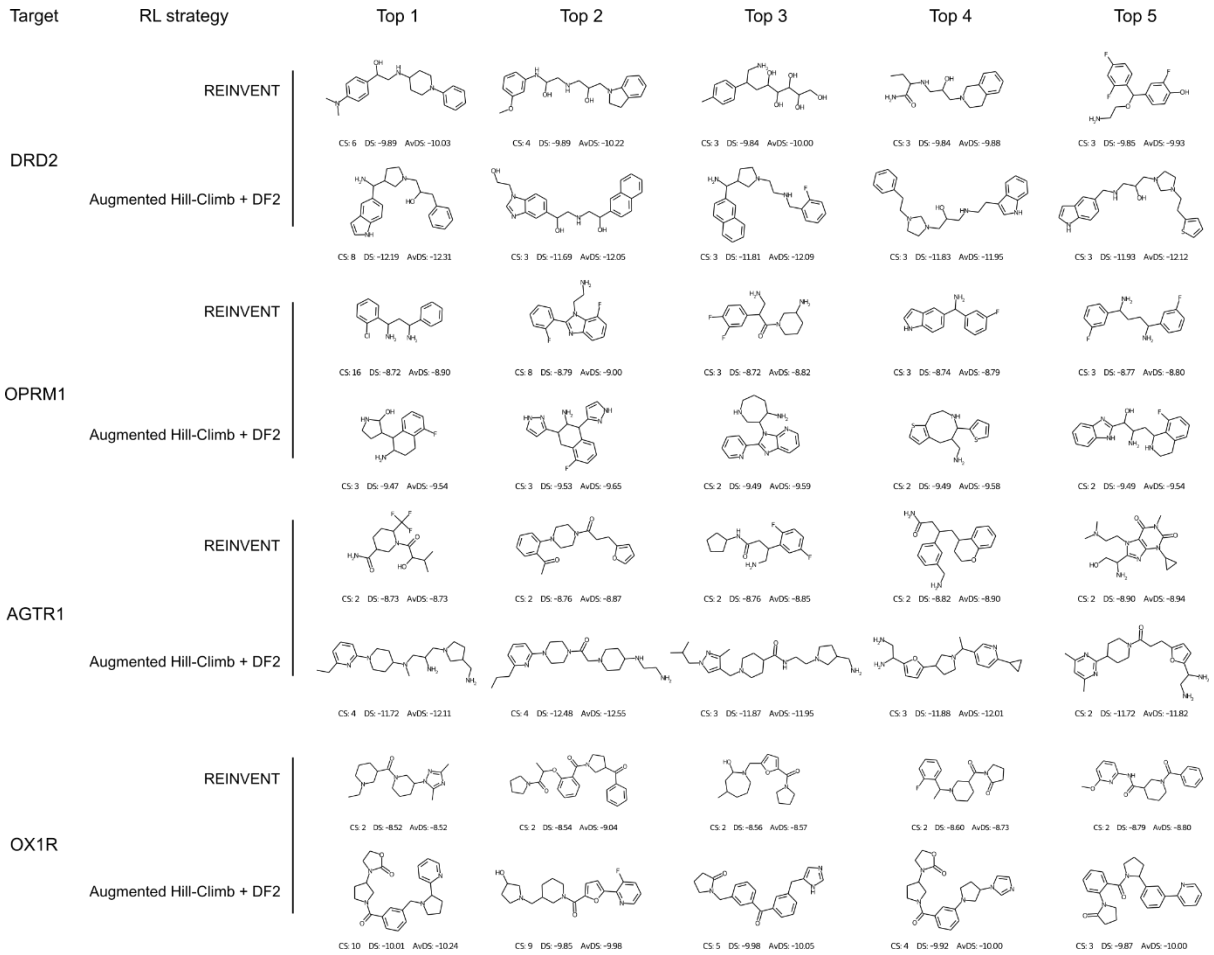

Figure 5: Centroid of the 5 largest clusters for the top 100 molecules according to docking score against DRD2, OPRM1, AGTR1 and OX1R receptors. Cluster size (CS), centroid docking score (DS) and the average cluster docking score (AvDS) is annotated below. In each case Augmented Hill-Climb generates clusters with lower (better) docking scores, while maintaining reasonable chemotypes that are indistinguishable to those generated by REINVENT. Note that protonation states, tautomers and stereoisomers are enumerated by the docking protocol (see Methods).

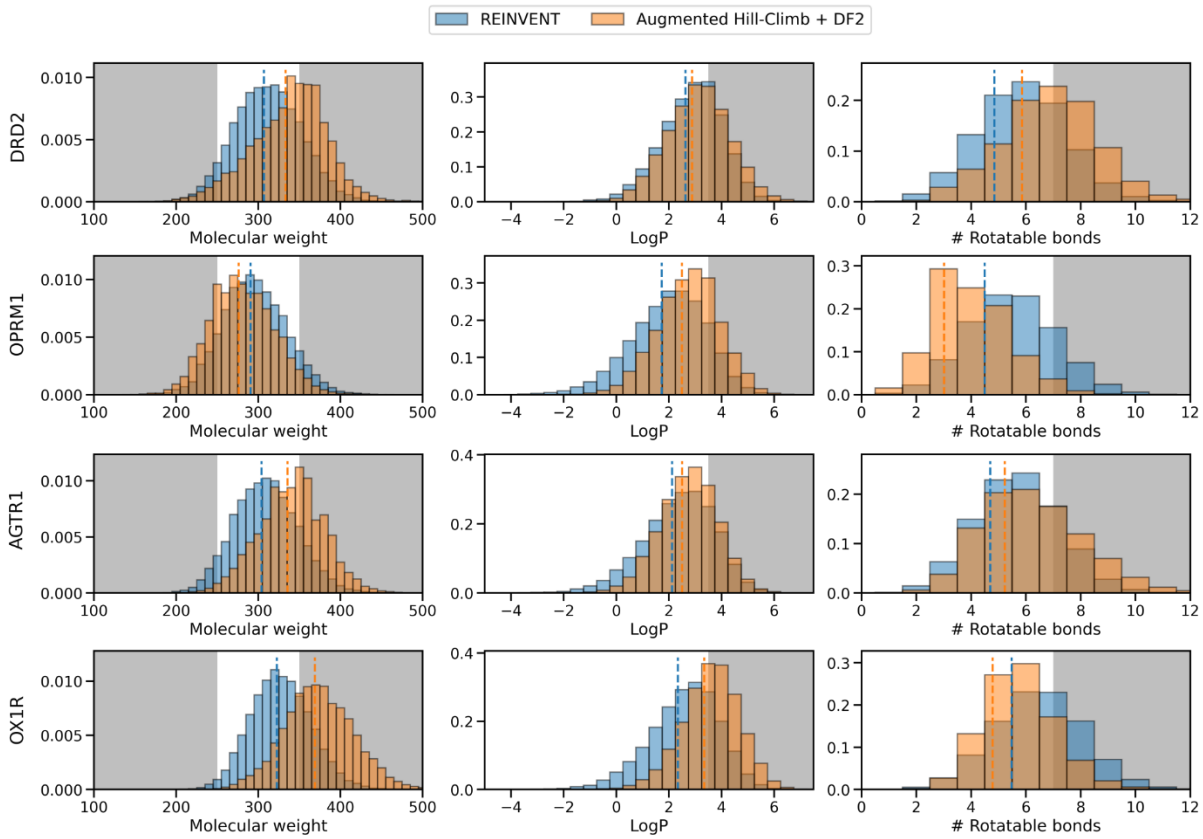

Figure 6: REINVENT compared to Augmented Hill-Climb (in combination with DF2) property space according to molecular weight, LogP and the number of rotatable bonds for molecules optimized to minimize the docking score against four targets. The grey shading indicates property space not represented in the prior training set.

Benchmarking Augmented Hill-Climb against other reinforcement learning strategies

The performance of Augmented Hill-Climb was compared to other RL strategies commonly used for language-based RNN de novo molecule generation, namely, REINFORCE [5], REINVENT [4, 53], BAR [19] and Hill-Climb [21], as well as in combination with KL regularization for non-regularized strategies (Experiment 4, Figure 1). In the interest of standardisation, the prior was trained on the GuacaMol train dataset. The RL strategies were applied to six tasks of varying difficulty (see Methods). DF2 was used in all cases except for the Risperidone similarity task which uses a lower minimum score threshold due (DF3) to low similarity values.

The performance of task optimization is shown in Figure 7. AHC is the most efficient of all RL strategies at all tasks except for maximizing the number of heavy atoms (Figure 7a). It is

particularly better than the other RL strategies during early-stage optimization (e.g., Figure 7b) and in more difficult objectives (e.g., Figure 7e, f). AHC even outperforms un-regularized RL strategies. This observation is true also for performance by wall time (Figure S9), a more practical measure. Intriguingly, AHC seems to achieve maximization towards the end of training in the heavy atom task (seen to a lesser extent with REINVENT *2.0*), suggesting it will eventually be able to extrapolate outside the training domain. As AHC uses a considerably smaller batch size than HC and therefore undergoes more frequent network updates, we applied the same batch size to HC to investigate this effect, denoted as HC*. This smaller batch size did in-fact improve sample-efficiency, similar to AHC*,* in early stages of training, but then quickly underwent mode collapse as evidenced by a drop in validity and uniqueness (Figures S10 and S11). Moreover, KL regularization did not rescue mode collapse in any case, and sometimes worsened performance, suggesting it is not a sufficient regularization method in this context. Interestingly, our re-implementation of BAR performed particularly poorly in most cases except for DRD2 activity (the case study in the original implementation [92]). We propose that the best agent memory in this method may actually inhibit learning without notable improvements in-between updating the 'best agent'; in effect having two 'regularizers' inhibiting learning. As a result, decreasing the 'best agent' update frequency (from 5 as originally implemented) may improve performance. Overall, AHC shows a sample-efficiency well beyond other RL strategies for all tasks of practical importance (i.e., excluding the heavy atom task).

Figures S12-17 show the centroids of the largest clusters for the top 100 molecules generated during the six benchmark optimization tasks. Firstly, all strategies are more prone to generating unrealistic chemistry due to the broader training domain of the GuacaMol [21] training set e.g., increasing molecular weight seen in the DRD2 docking score optimization task (Figure S15). This is even observed for the more heavily regularized REINVENT strategy but is not present when using the MOSES^{n} training set (Figure 5). Moreover, KL regularization as proposed previously [36, 66] does not seem to improve chemistry

generated by REINFORCE and HC and instead shows a tendency to increase molecular weight (Figure S14). On the other hand, AHC results in chemistry similar to REINVENT and is typically more reasonable than REINVENT 2.0 (e.g., longer linker chains in Figure S16), is less prone to idiosyncratic tendencies of HC (e.g., large molecules and long chains in Figure S16), yet more sample-efficient than either. Overall, we believe AHC strikes the right balance in the trade-off between extrapolation and sample-efficiency due to effective, tunable regularization that can maintain training set properties and therefore the generation of sensible and realistic molecules *de novo*.

We also acknowledge that other 'tricks' can be used to improve the sample-efficiency of RL. For example, experience replay can be used to remind the agent of 'good' molecules [53, 93] or a margin guard [94] can be employed to dynamically change α durin RL updates. We believe AHC is a more direct, principled approach to improve sample-efficiency and could even be used in combination with these tricks to further improve sample-efficiency.

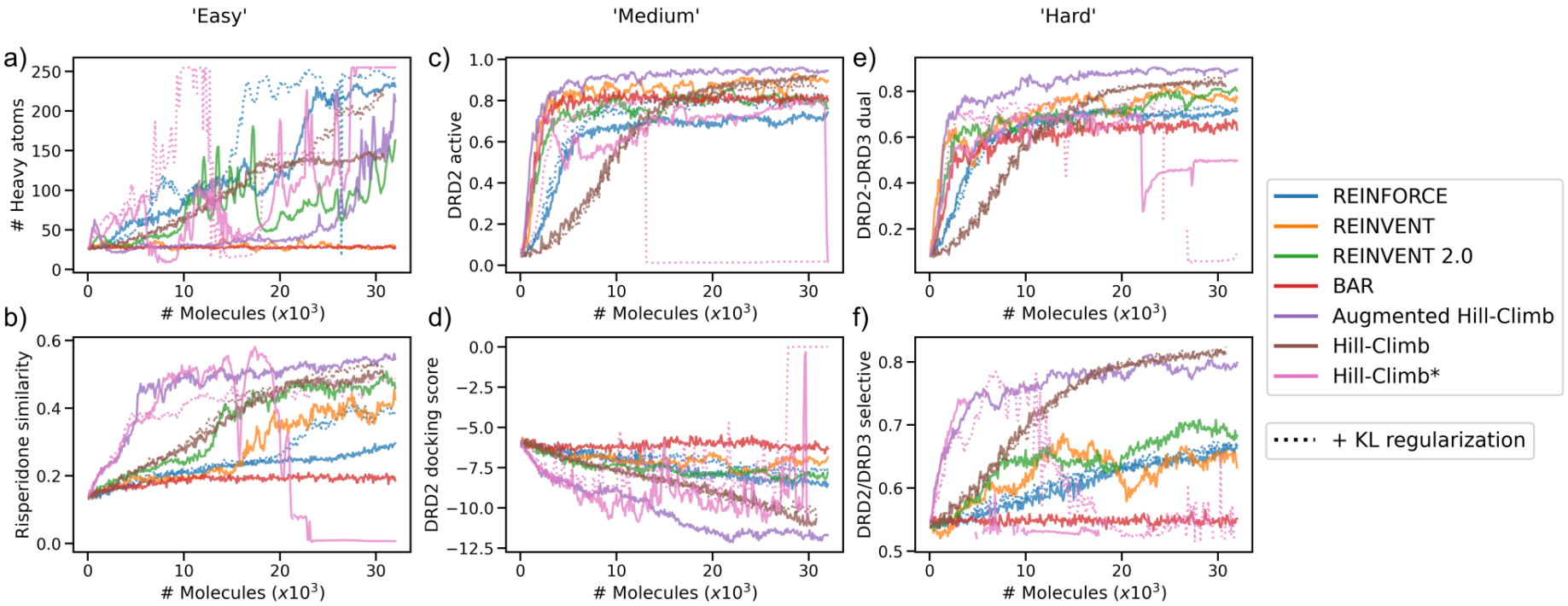

Figure 7: Per-molecule optimization of different RL strategies against different objective tasks of varying difficulty: (**a**) number of heavy atoms, (**b**) Similarity to Risperidone (DRD2 inverse agonist), (**c**) predicted probability of DRD2 activity, (**d**) Glide-SP docking score against DRD2, (**e**) predicted probability of dual activity against DRD2 and (**f**) predicted probability of selective activity towards DRD2 over DRD3. In all cases, except the number of heavy atoms, AHC outperforms all other RL strategies with respect to objective optimization while maintaining validity and uniqueness. Only valid molecules are plotted, therefore gaps seen with HC* denote regions where no valid molecules were generated.

Conclusion

In this work, we have proposed a modification to the REINVENT [4, 53] RL framework for language-based RNN de novo molecule generation that exhibits improved sample-efficiency. This method, referred to as Augmented Hill-Climb, improves optimization ability ~1.5-fold over REINVENT for the task of optimizing DRD2 Glide-SP [81] docking score. While more susceptible to mode collapse, this can be successfully ameliorated by application of an appropriate diversity filter. This new strategy can optimize the docking score for other systems beyond DRD2 including OPRM1, AGTR1 and OX1R where it improved sample-efficiency ~45-fold on average. When compared to other common RL strategies used in language-based RNN de novo molecule generation [5, 21, 36], it was found to outperform REINFORCE, REINVENT, BAR and Hill-Climb with respect to optimization ability, sample-efficiency, regularization and resulted in chemically reasonable molecules. We believe this is achieved by circumventing unwarranted regularization in REINVENT, but it can also be viewed as applying essential regularization to Hill-Climb. The improvement in sample-efficiency enabled by Augmented Hill-Climb will be especially useful when using computationally expensive scoring functions such as molecular docking or computer-aided synthesis planning tools. We believe these results highlight there is still scope for improvement in early generation ML-based generative models and that designing more complex generative models is not the only path to advance the field of molecular de novo design.

Abbreviations

ADAM: Adaptative moment estimation; AGTR1: AT_1 receptor; AHC: Augmented Hill-Climb; AI: Artificial Intelligence; BAR: Best agent reminder; DF: Diversity filter; DRD2: D_2 receptor D2; GPCR: ECFP: Extended-connectivity fingerprint; G protein-coupled receptor; GRU: Gated recurrent unit; HC: Hill-Climb; KL: Kullback-Leibler; LSTM: Long short-term memory; ML: Machine learning; MPO: Multi-parameter optimization; OPRM1: μ receptor; OX1R: OX_1

receptor; QSAR: Quantitative structure-activity relationship; RL: Reinforcement learning; RNN: Recurrent neural network; SMILES: Simplified molecular-input line-entry system.

Declarations

Availability of data and materials

The datasets supporting the conclusions of this article can be found via the following link https://doi.org/10.6084/m9.figshare.19591024.v1. The generative model code and scoring function code used to obtain the results discussed within the article is available at https://github.com/MorganCThomas/SMILES-RNN and https://github.com/MorganCThomas/MolScore, respectively.

Competing interests

The authors declare that they have no competing interests.

Funding

Morgan Thomas is funded by Sosei Heptares.

Authors' contributions

MT conducted this work under the supervision of AB and CDG with additional guidance from NO. The manuscript was revised and approved by all authors.

Acknowledgements

The authors acknowledge open-source tools used in this work. The permission to publish this work was granted by Sosei Heptares.

Additional files

Additional file 1 (.pdf): Supplementary tables and figures

References

1. Brown N, McKay B, Gilardoni F, Gasteiger J (2004) A graph-based genetic algorithm and its application to the multiobjective evolution of median molecules. J. Chem. Inf. Comput. Sci. 44:1079–1087

2. Jensen JH (2019) A graph-based genetic algorithm and generative model/Monte Carlo tree search for the exploration of chemical space. Chem Sci 10:3567–3572. https://doi.org/10.1039/C8SC05372C

3. Segler MHS, Kogej T, Tyrchan C, Waller MP (2018) Generating focused molecule libraries for drug discovery with recurrent neural networks. ACS Cent Sci 4:120–131. https://doi.org/10.1021/acscentsci.7b00512

4. Olivecrona M, Blaschke T, Engkvist O, Chen H (2017) Molecular de-novo design through deep reinforcement learning. J Cheminform 9:48. https://doi.org/10.1186/s13321-017-0235-x

5. Popova M, Isayev O, Tropsha A (2018) Deep reinforcement learning for de novo drug design. Sci Adv 4:. https://doi.org/10.1126/sciadv.aap7885

6. Yang X, Zhang J, Yoshizoe K, et al (2017) ChemTS: an efficient python library for de novo molecular generation. Sci Technol Adv Mater 18:972. https://doi.org/10.1080/14686996.2017.1401424

7. Moret M, Friedrich L, Grisoni F, et al (2020) Generative molecular design in low data regimes. Nat Mach Intell 2:171–180. https://doi.org/10.1038/s42256-020-0160-y

8. He J, You H, Sandström E, et al (2021) Molecular optimization by capturing chemist's intuition using deep neural networks. J Cheminform 13:. https://doi.org/10.1186/s13321-021-00497-0

9. Wang J, Hsieh C-Y, Wang M, et al (2021) Multi-constraint molecular generation based on conditional transformer, knowledge distillation and reinforcement learning. Nat

763 Mach Intell 2021 310 3:914–922. https://doi.org/10.1038/s42256-021-00403-1

764 10. Bagal V, Aggarwal R, Vinod PK, Priyakumar UD (2021) MolGPT: Molecular
765 Generation Using a Transformer-Decoder Model. J Chem Inf Model
766 acs.jcim.1c00600. https://doi.org/10.1021/ACS.JCIM.1C00600

767 11. Gómez-Bombarelli R, Wei JN, Duvenaud D, et al (2018) Automatic Chemical Design
768 Using a Data-Driven Continuous Representation of Molecules. ACS Cent Sci 4:268–
769 276. https://doi.org/10.1021/acscentsci.7b00572

770 12. Jin W, Barzilay R, Jaakkola T (2018) Junction tree variational autoencoder for
771 molecular graph generation. arXiv

772 13. Kajino H (2018) Molecular Hypergraph Grammar with its Application to Molecular
773 Optimization. arXiv

774 14. Winter R, Montanari F, Steffen A, et al (2019) Efficient multi-objective molecular
775 optimization in a continuous latent space. Chem Sci 10:8016–8024.
776 https://doi.org/10.1039/C9SC01928F

777 15. De Cao N, Kipf T (2018) MolGAN: An implicit generative model for small molecular
778 graphs. arXiv

779 16. Guimaraes GL, Sanchez-Lengeling B, Outeiral C, et al (2017) Objective-Reinforced
780 Generative Adversarial Networks (ORGAN) for Sequence Generation Models. arXiv

781 17. Blanchard AE, Stanley C, Bhowmik D (2021) Using GANs with adaptive training data
782 to search for new molecules. J Cheminform 13:14. https://doi.org/10.1186/s13321-
783 021-00494-3

784 18. Mercado R, Rastemo T, Lindelof E, et al (2021) Graph networks for molecular design.
785 Mach Learn Sci Technol 2:. https://doi.org/10.1088/2632-2153/abcf91

786 19. Atance SR, Diez JV, Engkvist O, et al (2021) De novo drug design using

reinforcement learning with graph-based deep generative models. ChemRxiv

20. Zhou Z, Kearnes S, Li L, et al (2019) Optimization of Molecules via Deep Reinforcement Learning. Sci Rep 9:10752. https://doi.org/10.1038/s41598-019-47148-x

21. Brown N, Fiscato M, Segler MHS, Vaucher AC (2019) GuacaMol: Benchmarking Models for de Novo Molecular Design. J Chem Inf Model 59:1096–1108. https://doi.org/10.1021/acs.jcim.8b00839

22. Polykovskiy D, Zhebrak A, Sanchez-Lengeling B, et al (2020) Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models. Front Pharmacol 11:1931. https://doi.org/10.3389/fphar.2020.565644

23. Popova M, Shvets M, Oliva J, Isayev O (2019) MolecularRNN: Generating realistic molecular graphs with optimized properties. arXiv

24. Thomas M, Boardman A, Garcia-Ortegon M, et al (2022) Applications of Artificial Intelligence in Drug Design: Opportunities and Challenges. Methods Mol Biol 2390:1–59. https://doi.org/10.1007/978-1-0716-1787-8_1

25. Weininger D (1988) SMILES, a Chemical Language and Information System: 1: Introduction to Methodology and Encoding Rules. J Chem Inf Comput Sci 28:31–36. https://doi.org/10.1021/ci00057a005

26. Jozefowicz R, Vinyals O, Schuster M, et al (2016) Exploring the Limits of Language Modeling. arXiv

27. Graves A, Eck D, Beringer N, Schmidhuber J (2004) Biologically plausible speech recognition with LSTM neural nets. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics) 3141:127–136. https://doi.org/10.1007/978-3-540-27835-1_10

28. Liu X, Ye K, van Vlijmen HWT, et al (2021) DrugEx v2: de novo design of drug

molecules by Pareto-based multi-objective reinforcement learning in polypharmacology. J Cheminformatics 2021 131 13:1–15. https://doi.org/10.1186/S13321-021-00561-9

29. Zhavoronkov A, Ivanenkov YA, Aliper A, et al (2019) Deep learning enables rapid identification of potent DDR1 kinase inhibitors. Nat Biotechnol 37:1038–1040. https://doi.org/10.1038/s41587-019-0224-x

30. Mercado R, Bjerrum EJ, Engkvist O (2021) Exploring Graph Traversal Algorithms in Graph-Based Molecular Generation. J Chem Inf Model. https://doi.org/10.1021/acs.jcim.1c00777

31. Moret M, Helmstädter M, Grisoni F, et al (2021) Beam Search for Automated Design and Scoring of Novel ROR Ligands with Machine Intelligence**. Angew Chemie - Int Ed 60:19477–19482. https://doi.org/10.1002/anie.202104405

32. Zhang J, Mercado R, Engkvist O, Chen H (2021) Comparative Study of Deep Generative Models on Chemical Space Coverage. J Chem Inf Model 61:2572–2581. https://doi.org/10.1021/ACS.JCIM.0C01328/SUPPL_FILE/CI0C01328_SI_001.PDF

33. Flam-Shepherd D, Zhu K, Aspuru-Guzik A (2021) Keeping it Simple: Language Models can learn Complex Molecular Distributions. arXiv

34. Cieplinski T, Danel T, Podlewska S, Jastrzębski S (2020) We Should At Least Be Able To Design Molecules That Dock Well. arXiv

35. Huang K, Fu T, Gao W, et al (2021) Therapeutics Data Commons: Machine Learning Datasets and Tasks for Drug Discovery and Development. arXiv

36. Neil D, Segler M, Guasch L, et al (2018) Exploring Deep Recurrent Models with Reinforcement Learning for Molecule Design. In: 6th International Conference on Learning Representations

37. Sutton RS, Barto AG (2018) Policy Gradient Methods. In: Reinforcement Learning: An

837 Introduction, 2nd ed. MIT Press, p 326

838 38. Tashiro M, Imamura Y, Katouda M (2020) De novo generation of optically active small
839 organic molecules using Monte Carlo tree search combined with recurrent neural
840 network. J Comput Chem jcc.26441. https://doi.org/10.1002/jcc.26441

841 39. Erikawa D, Yasuo N, Sekijima M (2021) MERMAID: an open source automated hit-to-
842 lead method based on deep reinforcement learning. J Cheminformatics 2021 131
843 13:1–10. https://doi.org/10.1186/S13321-021-00572-6

844 40. Boitreaud J, Mallet V, Oliver C, Waldispuhl J (2020) OptiMol: Optimization of binding
845 affinities in chemical space for drug discovery. J Chem Inf Model.
846 https://doi.org/10.1021/acs.jcim.0c00833

847 41. Jeon W, Kim D (2020) Autonomous molecule generation using reinforcement learning
848 and docking to develop potential novel inhibitors. Sci Rep 10:.
849 https://doi.org/10.1038/s41598-020-78537-2

850 42. Steinmann C, Jensen JH (2021) Using a genetic algorithm to find molecules with
851 good docking scores. PeerJ Phys Chem 3:. https://doi.org/10.7717/peerj-pchem.18

852 43. Nigam A, Pollice R, Krenn M, et al (2021) Beyond generative models: Superfast
853 traversal, optimization, novelty, exploration and discovery (STONED) algorithm for
854 molecules using SELFIES. Chem Sci 12:7079–7090.
855 https://doi.org/10.1039/d1sc00231g

856 44. Nigam A, Pollice R, Aspuru-Guzik A JANUS: Parallel Tempered Genetic Algorithm
857 Guided by Deep Neural Networks for Inverse Molecular Design

858 45. Xu Z, Wauchope OR, Frank AT (2021) Navigating Chemical Space by Interfacing
859 Generative Artificial Intelligence and Molecular Docking. J Chem Inf Model
860 10:acs.jcim.1c00746. https://doi.org/10.1021/ACS.JCIM.1C00746

861 46. Ma B, Terayama K, Matsumoto S, et al (2021) Structure-Based de Novo Molecular

Generator Combined with Artificial Intelligence and Docking Simulations. J Chem Inf Model 61:3304–3313. https://doi.org/10.1021/ACS.JCIM.1C00679/SUPPL_FILE/CI1C00679_SI_001.PDF

47. Thomas M, Smith RT, O'Boyle NM, et al (2021) Comparison of structure- and ligand-based scoring functions for deep generative models: a GPCR case study. J Cheminform 13:39. https://doi.org/10.1186/s13321-021-00516-0

48. Guo J, Janet JP, Bauer MR, et al (2021) DockStream: a docking wrapper to enhance de novo molecular design. J Cheminformatics 2021 131 13:1–21. https://doi.org/10.1186/S13321-021-00563-7

49. Ghanakota P, Bos PH, Konze KD, et al (2020) Combining Cloud-Based Free-Energy Calculations, Synthetically Aware Enumerations, and Goal-Directed Generative Machine Learning for Rapid Large-Scale Chemical Exploration and Optimization. J Chem Inf Model 60:4311–4325. https://doi.org/10.1021/acs.jcim.0c00120

50. Krishnan SR, Bung N, Bulusu G, Roy A (2021) Accelerating de Novo Drug Design against Novel Proteins Using Deep Learning. J Chem Inf Model 61:621–630. https://doi.org/10.1021/acs.jcim.0c01060

51. Su M, Yang Q, Du Y, et al (2019) Comparative Assessment of Scoring Functions: The CASF-2016 Update. J Chem Inf Model 59:895–913. https://doi.org/10.1021/acs.jcim.8b00545

52. Mysinger MM, Carchia M, Irwin JJ, Shoichet BK (2012) Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking. J Med Chem 55:6582–6594. https://doi.org/10.1021/jm300687e

53. Blaschke T, Arús-Pous J, Chen H, et al (2020) REINVENT 2.0: An AI Tool for De Novo Drug Design. J Chem Inf Model acs.jcim.0c00915. https://doi.org/10.1021/acs.jcim.0c00915

54. Fialková V, Zhao J, Papadopoulos K, et al (2021) LibINVENT: Reaction-based Generative Scaffold Decoration for in Silico Library Design. J Chem Inf Model. https://doi.org/10.1021/ACS.JCIM.1C00469/SUPPL_FILE/CI1C00469_SI_001.PDF

55. Thomas M (2021) MolScore. In: GitHub. https://github.com/MorganCThomas/SMILES-RNN/. Accessed 28 Mar 2022

56. Sterling T, Irwin JJ (2015) ZINC 15 - Ligand Discovery for Everyone. J Chem Inf Model 55:2324–2337. https://doi.org/10.1021/acs.jcim.5b00559

57. Wang R, Fu Y, Lai L (1997) A new atom-additive method for calculating partition coefficients. J Chem Inf Comput Sci 37:615–621. https://doi.org/10.1021/ci960169p

58. Kalgutkar AS, Soglia JR (2005) Minimising the potential for metabolic activation in drug discovery. Expert Opin Drug Metab Toxicol 1:91–142. https://doi.org/10.1517/17425255.1.1.91

59. Kalgutkar A, Gardner I, Obach R, et al (2005) A Comprehensive Listing of Bioactivation Pathways of Organic Functional Groups. Curr Drug Metab 6:161–225. https://doi.org/10.2174/1389200054021799

60. Baell JB, Holloway GA (2010) New substructure filters for removal of pan assay interference compounds (PAINS) from screening libraries and for their exclusion in bioassays. J Med Chem 53:2719–2740. https://doi.org/10.1021/jm901137j

61. Moret M, Friedrich L, Grisoni F, et al (2019) Generating customized compound libraries for drug discovery with machine intelligence. ChemRxiv. https://doi.org/10.26434/CHEMRXIV.10119299.V1

62. BenevolentAI GuacaMol Baselines. In: GitHub. https://github.com/BenevolentAI/guacamol_baselines. Accessed 3 Mar 2022

63. Sutton RS, Barto AG (2018) Reinforcement Learning: an introduction, second edi. MIT Press

64. Williams RJ (1992) Simple statistical gradient-following algorithms for connectionist reinforcement learning. Mach Learn 8:229–256. https://doi.org/10.1007/bf00992696

65. Popova M, Isayev O, Tropsha A (2017) Deep Reinforcement Learning for De-Novo Drug Design. https://doi.org/10.1126/sciadv.aap7885

66. Jaques N, Gu S, Bahdanau D, et al (2016) Sequence Tutor: Conservative Fine-Tuning of Sequence Generation Models with KL-control. 34th Int Conf Mach Learn ICML 2017 4:2587–2596

67. Blaschke T, Engkvist O, Bajorath J, Chen H (2020) Memory-assisted reinforcement learning for diverse molecular de novo design. J Cheminform 12:68. https://doi.org/10.1186/s13321-020-00473-0

68. Rogers D, Hahn M (2010) Extended-connectivity fingerprints. J Chem Inf Model 50:742–754. https://doi.org/10.1021/ci100050t

69. Bemis GW, Murcko MA (1996) The properties of known drugs. 1. Molecular frameworks. J Med Chem 39:2887–2893. https://doi.org/10.1021/jm9602928

70. Smith DH, Carhart RE, Venkataraghavan R (1985) Atom Pairs as Molecular Features in Structure-Activity Studies: Definition and Applications. J Chem Inf Comput Sci 25:64–73. https://doi.org/10.1021/ci00046a002

71. RDKit Open-source cheminformatics. http://www.rdkit.org

72. Wang S, Che T, Levit A, et al (2018) Structure of the D2 dopamine receptor bound to the atypical antipsychotic drug risperidone. Nature 555:269–273. https://doi.org/10.1038/nature25758

73. Manglik A, Kruse AC, Kobilka TS, et al (2012) Crystal structure of the μ-opioid receptor bound to a morphinan antagonist. Nat 2012 4857398 485:321–326. https://doi.org/10.1038/nature10954

936 74. Zhang H, Unal H, Gati C, et al (2015) Structure of the angiotensin receptor revealed
937 by serial femtosecond crystallography. Cell 161:833–844.
938 https://doi.org/10.1016/J.CELL.2015.04.011/ATTACHMENT/E73AA1EA-1C95-4167-
939 9AF7-32BCCD019647/MMC1.PDF

940 75. Rappas M, Ali AAE, Bennett KA, et al (2020) Comparison of Orexin 1 and Orexin 2
941 Ligand Binding Modes Using X-ray Crystallography and Computational Analysis. J
942 Med Chem 63:1528–1543. https://doi.org/10.1021/acs.jmedchem.9b01787

943 76. Schrödinger Release 2019-4 Protein Preparation Wizard

944 77. Shelley JC, Cholleti A, Frye LL, et al (2007) Epik: A software program for pKa
945 prediction and protonation state generation for drug-like molecules. J Comput Aided
946 Mol Des 21:681–691. https://doi.org/10.1007/s10822-007-9133-z

947 78. Sondergaard CR, Olsson MHM, Rostkowski M, Jensen JH (2011) Improved treatment
948 of ligands and coupling effects in empirical calculation and rationalization of p K a
949 values. J Chem Theory Comput 7:2284–2295. https://doi.org/10.1021/ct200133y

950 79. Roos K, Wu C, Damm W, et al (2019) OPLS3e: Extending Force Field Coverage for
951 Drug-Like Small Molecules. J Chem Theory Comput 15:1863–1874.
952 https://doi.org/10.1021/acs.jctc.8b01026

953 80. Schrödinger Release 2019-4 LigPrep

954 81. Friesner RA, Banks JL, Murphy RB, et al (2004) Glide: A New Approach for Rapid,
955 Accurate Docking and Scoring. 1. Method and Assessment of Docking Accuracy. J
956 Med Chem 47:1739–1749. https://doi.org/10.1021/jm0306430

957 82. Sun J, Jeliazkova N, Chupakhin V, et al (2017) ExCAPE-DB: an integrated large scale
958 dataset facilitating Big Data analysis in chemogenomics. J Cheminform 9:17.
959 https://doi.org/10.1186/s13321-017-0203-5

960 83. Ashton M, Barnard J, Casset F, et al (2002) Identification of Diverse Database

961 Subsets using Property-Based and Fragment-Based Molecular Descriptions. Quant
962 Struct Relationships 21:598–604. https://doi.org/10.1002/QSAR.200290002

963 84. Pedregosa F, Varoquaux G, Gramfort A, et al (2011) Scikit-learn: Machine Learning in
964 Python. J Mach Learn Res 12:2825–2830

965 85. Esposito C, Landrum GA, Schneider N, et al (2021) GHOST: Adjusting the Decision
966 Threshold to Handle Imbalanced Data in Machine Learning. J Chem Inf Model
967 acs.jcim.1c00160. https://doi.org/10.1021/acs.jcim.1c00160

968 86. Pan Y, Huang N, Cho S, MacKerell AD (2003) Consideration of molecular weight
969 during compound selection in virtual target-based database screening. J Chem Inf
970 Comput Sci 43:267–272. https://doi.org/10.1021/ci020055f

971 87. Preuer K, Renz P, Unterthiner T, et al (2018) Fréchet ChemNet Distance: A Metric for
972 Generative Models for Molecules in Drug Discovery. J Chem Inf Model 58:1736–
973 1741. https://doi.org/10.1021/acs.jcim.8b00234

974 88. Arús-Pous J, Johansson SV, Prykhodko O, et al (2019) Randomized SMILES strings
975 improve the quality of molecular generative models. J Cheminform 11:71.
976 https://doi.org/10.1186/s13321-019-0393-0

977 89. Vass M, Podlewska S, De Esch IJP, et al (2019) Aminergic GPCR-Ligand
978 Interactions: A Chemical and Structural Map of Receptor Mutation Data. J. Med.
979 Chem. 62:3784–3839

980 90. Vass M, Kooistra AJ, Ritschel T, et al (2016) Molecular interaction fingerprint
981 approaches for GPCR drug discovery. Curr Opin Pharmacol 30:59–68.
982 https://doi.org/10.1016/j.coph.2016.07.007

983 91. Kaczor AA, Silva AG, Loza MI, et al (2016) Structure-Based Virtual Screening for
984 Dopamine D2 Receptor Ligands as Potential Antipsychotics. ChemMedChem
985 11:718–729. https://doi.org/10.1002/cmdc.201500599

92. Khemchandani Y, O'Hagan S, Samanta S, et al (2020) DeepGraphMolGen, a multi-objective, computational strategy for generating molecules with desirable properties: a graph convolution and reinforcement learning approach. J Cheminform 12:53. https://doi.org/10.1186/s13321-020-00454-3

93. Korshunova M, Huang N, Capuzzi S, et al (2021) A Bag of Tricks for Automated De Novo Design of Molecules with the Desired Properties: Application to EGFR Inhibitor Discovery. ChemRxiv. https://doi.org/10.26434/CHEMRXIV.14045072.V1

94. Patronov A, Margreitter C, Blaschke T, Guo J (2021) REINVENT 3.0. In: GitHub. https://github.com/MolecularAI/Reinvent/tree/reinvent.3.0. Accessed 28 Mar 2022