Sample Efficient Reinforcement Learning with Active Learning for Molecular Design

Michael Dodds[1], Jeff Guo[1*], Thomas Löhr[1], Alessandro Tibo[1], Ola Engkvist[1], Jon Paul Janet[1]

1: Molecular AI, Discovery Sciences, R&D, AstraZeneca, 431 50 Gothenburg, Sweden

Abstract

Reinforcement learning (RL) is a powerful and flexible paradigm for searching for solutions in high-dimensional action spaces. However, bridging the gap between playing computer games with thousands of simulated episodes and solving real scientific problems with complex and involved environments (up to actual laboratory experiments) requires improvements in terms of sample efficiency to make the most of expensive information. The discovery of new drugs is a major commercial application of RL, motivated by the very large nature of the chemical space and the need to perform multiparameter optimization (MPO) across different properties. *In silico* methods, such as virtual library screening (VS) and de-novo molecular generation with RL, show great promise in accelerating this search. However, incorporation of increasingly complex computational models in these workflows requires increasing sample efficiency. Here, we introduce an active learning system linked with an RL model (RL-AL) for molecular design, which aims to improve the sample-efficiency of the optimization process. We identity and characterize unique challenges combining RL and AL, investigate the interplay between the systems, and develop a novel AL approach to solve the MPO problem. Our approach greatly expedites the search for novel solutions relative to baseline-RL for simple ligand- and structure-based oracle functions, with a 1000-

75 000%-increase in hits generated for a fixed oracle budget and a 14-65-fold reduction in computational time to find a specific number of hits. Furthermore, compounds discovered through RL-AL display substantial enrichment of a multi-parameter scoring objective, indicating superior efficacy in curating high-scoring compounds, without a reduction in output diversity. This significant acceleration improves the feasibility of oracle functions that have largely been overlooked in RL due to high computational costs, for example free energy perturbation methods, and in principle is applicable to any RL domain.

Introduction

The computational design of molecules with specific profiles is a key scientific and technological challenge[1] across many important application areas from catalysis and energy storage, to the design of pharmaceutical drugs. This task is complicated by the very large size of chemical space[2], and the requirement to fulfil multiple design criteria (multiparameter optimization, MPO). In drug design, candidate molecules must be active against an intended target but also possess suitable physicochemical, metabolic and safety profiles. Despite advances in automated chemical synthesis, the scale of chemical space makes computational evaluation of candidate molecules essential for accelerating molecular discovery[3]. Traditional virtual screening (VS) involves exhaustively evaluating a large library of molecules (up to billions[4-6]) to identify candidates with the desired predicted properties, called hits, which comprise a small fraction of the total library that is screened.

A variety of computational models is available to assess hits in VS, from simple data-driven methods (quantitative-structure-activity relationships, QSAR) to physics-based computation via pharmacophore matching methods or molecular docking, whereby a putative binding pose of the molecule is generated and scored[7] for compatibility with a target protein. Such methods have been successfully applied to VS efforts[6,8,9], although the incorporation of docking already imposes a substantial computational burden (100s of *CPU-years in Ref.*[5]

and one of the largest supercomputers in the world in Ref.[6]) when screening large libraries. Recent developments of high-accuracy, high-computational-cost binding affinity prediction methods with molecular dynamics such as free-energy perturbation[10,11] (FEP) or non-equilibrium switching[12], have become the new gold-standard for affinity prediction[13], but are prohibitively computationally expensive and cannot be directly applied to large VS libraries.

Although an old idea[14,15], active learning (AL) methods have recently gained increasing attention for accelerating VS[16], either to enable screening very large libraries with docking[17], or screening smaller libraries with binding energy prediction[18-20] or quantum chemical methods[21-23]. VS-AL methods generically sample a small subset of compounds to evaluate with a desired scoring function, or *oracle*, and construct a surrogate model to predict the oracle score of as-yet unevaluated candidates. This model is used to select new candidates to screen, based on an *acquisition function* which might depend on the surrogate predictions, associated uncertainties and other factors. Evaluated molecules are used to retrain the surrogate model and the approach is iteratively repeated. Such approaches regularly claim hundred-fold or more accelerations over brute-force VS in terms of oracle calls needed to retrieve top-scoring hits.

As an alternative to traditional VS, deep generative methods have transitioned from research protypes to practical and powerful tools for computational drug design[24-26]. Such models are responsible for the design of multiple experimentally validated hits, including potent small molecule inhibitors for a variety of targets[27-32] and PROTACs[33]. Rather than screening a fixed, finite library, generative chemical models[34] propose novel molecules based on probabilistic principles, allowing them to address very large chemical design spaces[35]. These de novo design models consist of a generative component responsible for sampling molecules and a mechanism for steering the design to molecules with target properties. Existing methodologies for the generative component could include text- or graph-based variational autoencoders[36-38], generative adversarial networks[39,40], sequence/recurrent models[41-44], transformers[45-48] or diffusion models[49,50] , while the steering mechanism typically involves either conditional generation (i.e. on a target or profile) or an optimization method such as reinforcement learning (RL).

Here, we consider REINVENT[41,51], a SMILES[52]-based (Simplified Molecular Input Line Entry System) recurrent-neural network (RNN) based molecule generator that utilizes policy-gradient RL to iteratively improve suggested molecules according to a flexible scoring function that can include a variety of scoring components including docking[53,54] and ROCS[55]. MPO is achieved by normalizing each scoring component to a 0-1 range and averaging over scoring components. The relative simplicity and flexibility of REINVENT makes it a popular testbed for experiments with molecular RL[56-58].

One major concern with RL methods in the real-world is sample efficiency[59], i.e. the number of oracle calls needed to reliably learn the desired output. While REINVENT has been found to be among the most sample-efficient generative chemical models[60], it still typically requires thousands of oracle evaluations to learn to produce favorable molecules. While this may compare favorably with the costs of brute-force VS on large libraries, the incorporation of higher-cost simulations remains prohibitive. We recently introduced a curriculum-learning approach whereby a simpler, physically-motivated function is learned first, for example learning a ROCS query before starting docking, which can substantially reduce the number of expensive oracle evaluations[61]. However, this approach depends on identifying physically-motivated intermediate objectives that are correlated with the desired oracle.

Here, we instead investigate accelerating RL for molecular design in an oracle agnostic manner using AL (RL-AL). The RL-AL setting poses unique challenges for surrogate-model based AL relating to the inherent feedback loops in the generative RL setting. We begin with a motivating example that illustrates some of these unique difficulties and general implications of RL-AL compared to traditional AL. We then systematically examine the components of the RL-AL system and design a strategy that can accelerate generative molecular design by a ~23-fold reduction in oracle calls and a ~14-65-fold reduction in CPU time. Next, we introduce a new acquisition strategy that is compatible with the MPO nature of the RL process. Finally, we demonstrate the transferability of our approach across oracle functions and quantify the computational- and wall-time saving of our method.

Results

Motivating Example

We consider a VS setting to evaluate the implications of RL and AL methods from an oracle-call efficiency (novel hits per oracle call) perspective. We begin by sampling 100 000 molecules from the REINVENT prior, which is trained to mimic the ChEMBL[62] database, and evaluating the complete library with two distinct computational models/oracles (**Methods**):

1) a shape and color pharmacophore query for Cyclooxygenase-2 (COX2) (PDB ID: 1CX2) [63] based on the native SC-558 ligand implemented using ROCs[64]
2) a docking protocol for the Retinoic Acid Receptor Alpha (RXRα) (PDB ID: 7B88)[65], implemented using AutoDock-Vina[66]

We judge hits for each oracle based on having a greater predicted affinity than obtained for their respective native ligands (0.6 for ROCs and -11.4 kcal mol^{-1} for the docking oracle respectively). With exhaustive screening of the library, we identify 30 unique Murcko scaffolds[67] and 41 unique hit SMILES for ROCS (364 and 369 for docking), for a hit rate or oracle-call-efficiency of 0.03% / 0.04% (0.36% / 0.37% for docking).

Initially we test active learning for virtual screening (VS-AL). In agreement with previous studies[17] we obtain a substantial increase in oracle efficiency compared to brute-force screening, recovering 42±4.0 and 35±4% of hit molecules with only 5000 oracle calls (**Figure 1**), for a hit rate of 0.25% and 2.54% in 5000 oracle calls for ROCS and docking respectively, a 7-11 fold improvement. The VS-AL recovers all hits within 8927±1750 oracle calls for ROCS and 82% of hits (~314) after 30 000 oracle calls for docking.

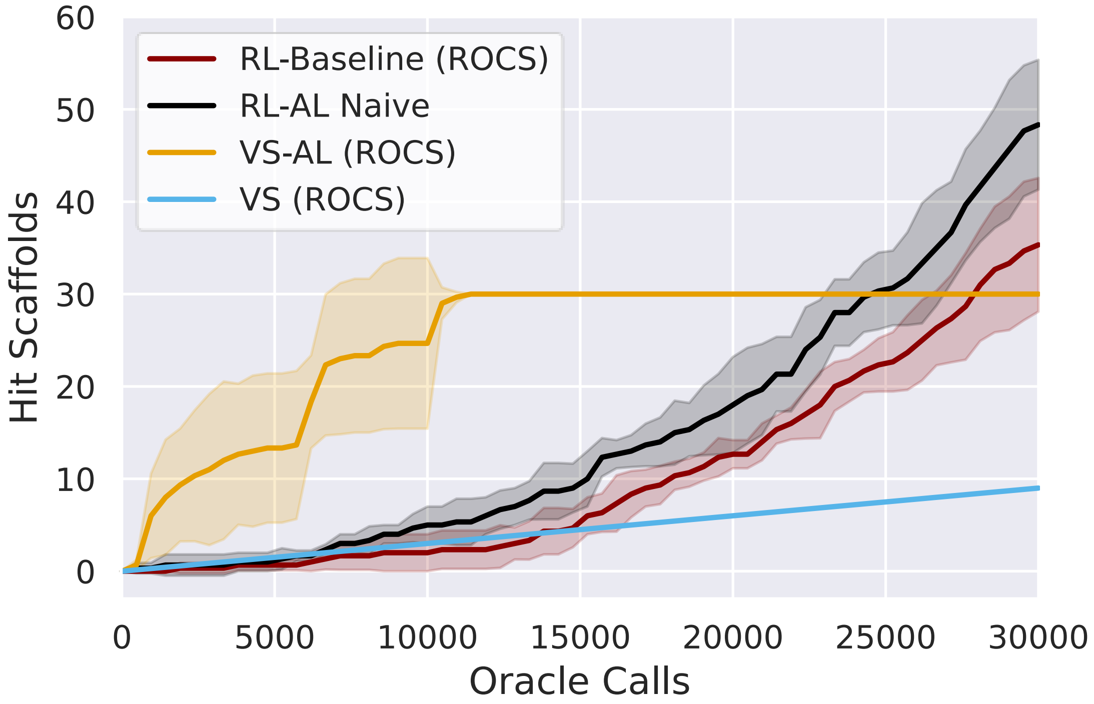

Figure 1: Comparison of VS, VS-AL, Baseline REINVENT (RL) and RL-AL., showing the number of hit scaffolds discovered using a ROCS oracle as a function of oracle calls for 30 000 total calls. Lines show the mean of three repeats of each experiment while the shading indicates one standard deviation. Hit finding is limited in VS-AL primarily due to exhaustion of hits at ~ 10 000 oracle calls. Generative models (RL, RL-AL)show exponential growth in the number of hits found between 10 000 and 30 000 oracle calls. By 30 000 calls, both RL systems have generated more hits than those contained in the virtual library (30 hits out of 100 000 total compounds).

Next, we compared VS-AL performance to a RL approach with REINVENT[41,51]. We used a standard RL configuration with a batch size of 128 (**Methods**) where we sought to improve the oracle score of the generated compounds along with a few commonly used metrics for molecule quality (**Methods**) to provide a realistic MPO setting. We performed each experiment in triplicate.

Initially, the hit rate of the RL agent is comparable to VS. However, after ~15 000 oracle calls, the RL agent learns to reliably generate hits, after which time the productivity of the generative model grows rapidly, and by 30 000 calls, the RL method has produced more unique hits (79±18) and scaffolds (35±7) for ROCs than contained in the virtual library (**Figure 1**). The final overall oracle call efficiency is 0.12% for ROCs. For docking, 37±8 hits are generated (all unique scaffolds), a lower number than contained in the virtual library, for a docking oracle efficiency of 0.12%. The compounds generated via RL exhibit higher average component scores due to MPO (**Supporting Figure 1**).

We added a naïve AL component into the RL loop based on the VS-AL model (Reinforcement Learning with Active Learning in Methods). In contrast to the immediate, rapid increase in hit rate observed in VS-AL, the RL-AL approach barely improves the hit rate obtained in the early phase and lead to moderate increases in total productivity by 30 000 epoch calls, resulting in 134±23 and 49±5 hits (48±7 and 47±5 scaffolds) and an oracle call efficiency of 0.16% for both ROCs and docking respectively. While an increase over the basic RL case, this is far from realizing the benefits observed in the VS-AL.

We identify some key factors that make RL-AL uniquely challenging:

1. **Non-stationary distribution:** As RL proceeds, the underlying generative model is updated, and the type of molecules generated in later epochs are distinct from the preceding steps (**Supporting Figure 2**). We observed that a surrogate model trained on fixed data will consistently lose predictive accuracy on later epochs **(Supporting Figure 3)**, limiting the utility of persisting data collected during the run.
2. **Feedback loops and robustness:** Because the scores produced by the surrogate model directly influence the molecules generated in the next epoch, incorrect scores in the early epochs interfere with the learning of the RL agent. To illustrate this, we ran RL but added Gaussian noise to the oracle and observed a noise-level-dependent decrease in learning efficiency **(Supporting Figure 4)**. Furthermore, REINVENT is already an iterative process of RL updates, and previous studies have established the sensitivity of the RL process to this update frequency[58]. Introducing AL creates a second internal loop, with additional hyperparameters relating to the relative frequency of the RL and AL updates.

In the following sections, we derive experiments to investigate various strategies for optimally leveraging the benefits of AL to accelerate RL and obtain drastically improved oracle call efficiency.

Experimental design

Here, we consider a set of experiments to investigate the interplay between AL and RL, and the impact of various parametric choices on this process. The overall RL-AL process consists of the following high-level steps (**Figure 2**):

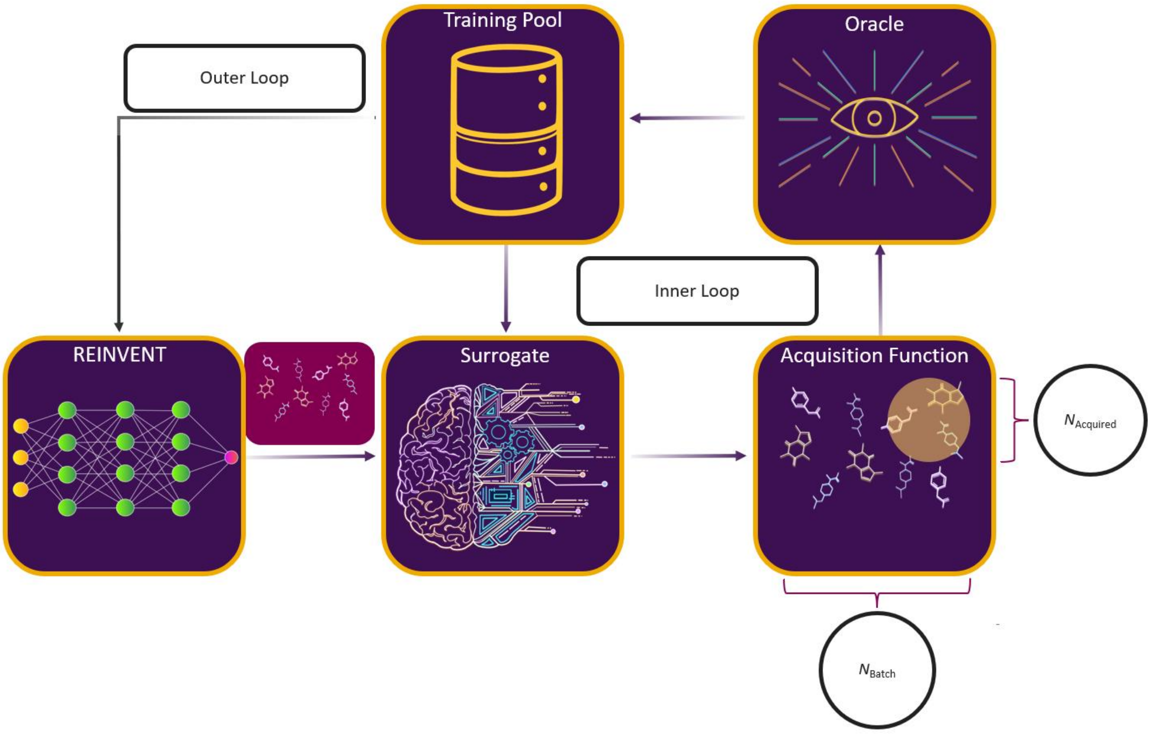

Figure 2: Schematic of an RL-AL system for drug design. REINVENT[51,57,68] generates drug-like compounds encoded as SMILES strings[52]. The generated SMILES are input to the surrogate model which predicts the oracle scores for each compound. Based on the specified acquisition function, a subset of compounds is sent for ground-truth label acquisition using the oracle function, while the non-acquired compounds use the surrogate-predicted labels. The oracle-labelled compounds are pooled and used to retrain the surrogate model. The predict, split, label, and train cycle is repeated for N_{loops} (inner loop). The combined set is then passed to back REINVENT for computing the appropriate RL update (the outer loop).

1) REINVENT generates N_{batch} compounds based on the current agent state.

2) The current surrogate model predicts oracle labels for each generated compound.

3) An acquisition function is used to select $\frac{N_{\text{acquired}}}{N_{\text{loops}}}$ compounds to evaluate with the oracle based on the surrogate predictions.
4) The surrogate model is updated based on oracle predictions for the most recent $N_{\text{training pool}}$ compounds in a sliding-window scheme.
5) Steps 3-4 are repeated N_{loops} times to acquire exactly N_{acquired} compounds per epoch.
6) The RL agent is updated based on the oracle generated labels where they exist and the surrogate predicted values otherwise, potentially with a different weighting.

Full details of this process are provided in **Methods**. Here, we investigate the impact of varying factors related to how the RL policy update is performed and how AL is conducted to balance between RL and AL updates.

As in the naïve RL-AL experiment, we use N_{batch} of 256 and N_{acquired} of 128 in one loop with UCB acquisition (**Methods**) as a baseline case. We use the ROCS oracle for experimentation due to its better computational efficiency compared to docking[64]. A summary of all configuration parameters is provided in the Supporting Information (**Supporting Table 1**). We test the efficiency of each configuration from the perspective of number of unique hit scaffolds, predicted affinity/overlay score greater than the native ligand, generated and acquired per oracle call over 30 000 oracle calls and the diversity of the resulting generated hit scaffolds (**Figure 3**).

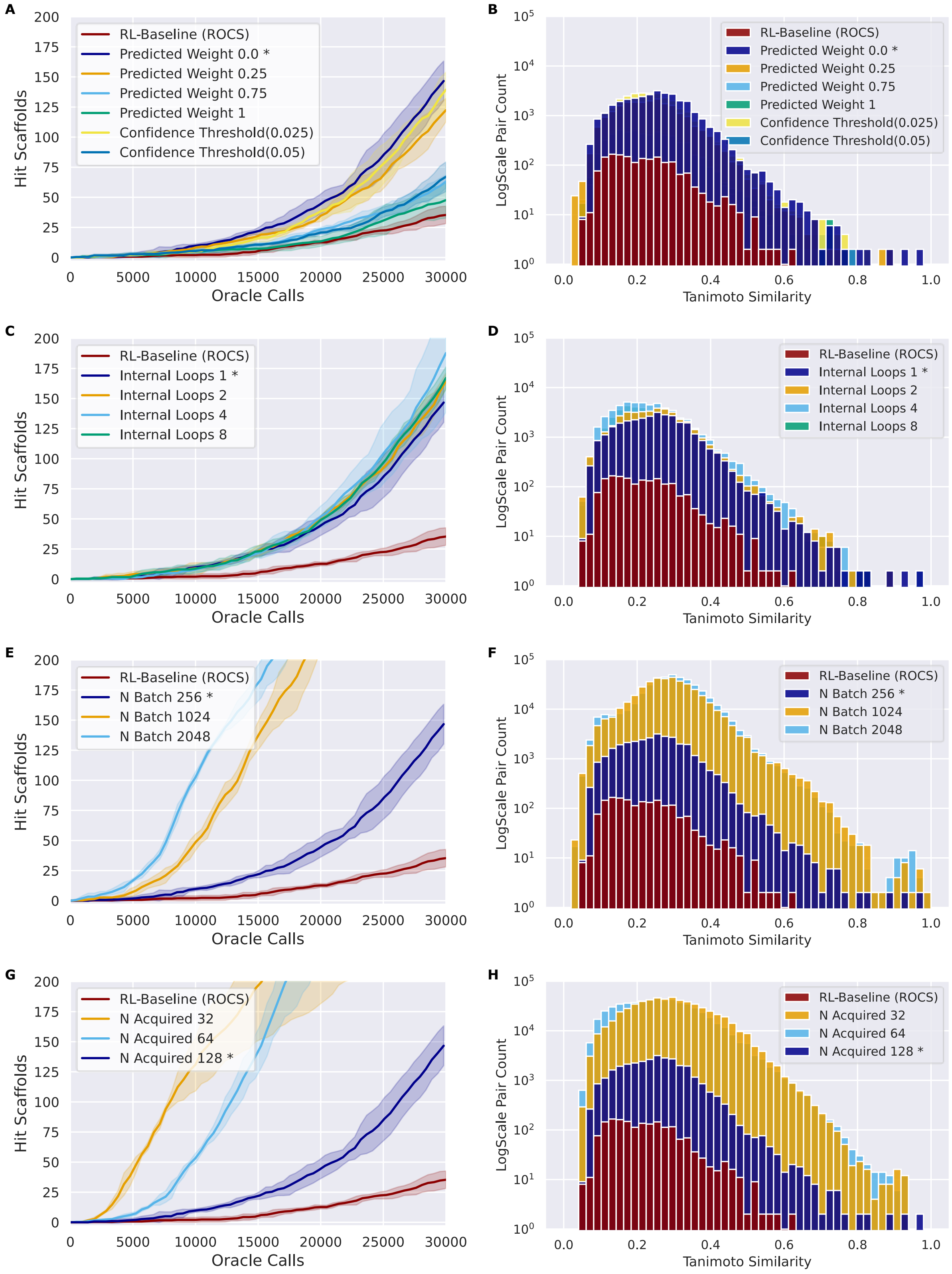

Figure 3: Composite comparing the impact of various design choices (rows) on the number of hit scaffolds obtained over 30 000 calls to a ROCS oracle function (left) and the average pairwise Tanimoto similarity based on ECFP4 fingerprints for the identified compounds (right). Lines show the mean of triplicate experiments while the shading indicates one standard deviation (left), and histograms show

the combined results of all replicates (right). The RL-AL baseline case, held constant across all trials, is marked with an asterisk.

Weights of the RL update

Here, we introduced a weighted loss function for the RL process (**Methods**) and use this to explore down-weighting updates based on surrogate predictions relative to oracle-acquired labels (always given a weight of 1) to counteract the sensitivity of the RL-AL process to errors in the early surrogate model (**Figure 3A/B**). We consider weighting the surrogate-prediction and oracle predictions equally (full belief), weighting the surrogate model predictions 0.75 or 0.25 to put more onus on oracle-predicted values, setting the weights for compounds to 0 where the surrogate model uncertainty is too high (> 0.025 or > 0.05) (**Supporting Figure 5**) ,and finally using a 0 weight for all surrogate predictions – thereby updating the RL agent based on the oracle only.

Compared to RL-AL with equal weights, we observe a substantial increase in the number of hit scaffolds generated when down-weighting surrogate model predictions. Despite reasonable predictive performance of the surrogate model (look-ahead mean average error, MAE, of 0.046 for the AL ROCS case), the RL-process with zero weights for surrogate prediction results in 147±14 hit scaffolds after 30 000 epochs vs 48±6 for the equal-weights RL-AL model (**Figure 3A**). The RL agent is not updated based on the surrogate predictions at all in this case; instead, the AL-subprocess is effectively curating high-scoring compounds for the RL update, focusing the RL learning on high-scoring compounds.

All RL-AL interventions show marginally lower hit scaffold diversity (**Figure 3B**), with an average pairwise similarity of 0.212±0.017 for the zero-weight update scheme vs 0.178±0.003 for the equal weights RL run. Since the impact of updating with zero-weights for surrogate compounds is so drastic, we perform all future experiments with this updating scheme (defined as RL-AL baseline, indicated by asterisks in **Figure 3**).

Acquisition strategy and surrogate model parameters

The acquisition function is used to select a subset of the generated batch, X_A, to evaluate with the oracle, e.g. $X_A \subset X;\ |X_A| = \frac{N_{\text{acquired}}}{N_{\text{loops}}} \leq N_{\text{batch}} = |X|$, though we also explore iterative construction of this set. We investigate and compare the performance of two distinct strategies: greedy acquisition and UCB (**Methods**). We selected these strategies due to their varying approaches to balancing exploration vs exploitation and success in previous studies[17]. Note that, in the context of the RL-AL with weight zero used for surrogate predictions model, random acquisition is equivalent to not using AL at all since no surrogate predictions are used for the RL update. UCB found 27% more scaffolds compared to greedy (116±8 for greedy vs 147±13 for UCB) with a similar Tanimoto diversity (0.205±0.005 for greedy vs 0.212±0.017 for UCB), both methods outperform random acquisition (**Supporting Figure 6**).

AL batch size, RL batch size and number of AL loops

We extensively evaluated the RL-AL relationship through varying the size of the REINVENT batch, N_{batch}, the number of compounds acquired in total, N_{acquired}, and the number of AL loops per RL epoch, N_{loops}. Intuitively, the larger the AL/RL ratio $\frac{N_{\text{acquired}}}{N_{\text{batch}}}$, the closer the result will be to baseline RL (ratio=1) and the lower the potential for reducing oracle calls.

1. First, with a fixed $N_{\text{batch}} = 256$ compounds and with a fixed total $N_{\text{acquired}} = 128$, we vary N_{loops} from 1 to 8, iteratively extending the training set and corresponding to a fixed AL/RL ratio $\frac{N_{\text{acquired}}}{N_{\text{batch}}} = 0.5$ (**Figure 3C/D**).
2. Second, we increase the size of N_{batch} up to 2048 compounds with $N_{\text{acquired}} = 128$ and $N_{\text{loops}} = 1$, representing AL/RL ratios from 0.0625 to 0.5 (**Figure 3E/F**).
3. Next, we vary N_{acquired} from 32 to 128 at fixed $N_{\text{batch}} = 256$ and $N_{\text{loops}} = 1$, corresponding to AL/RL ratios from 0.125 to 0.5 (**Figure 3G/H**).

4. Finally, we test the cross-dependence of these factors, by varying N_{acquired} in [32, 64,128], N_{batch} in [128, 256, 512, 1024, 2048], and N_{loops} in [1, 2, 4, 8] (**Figure 4**) (Full results available in (**Supporting Table 2**).

Comparing the impact of these parameters on the number of hit scaffolds generated (**Figure 3C/E/G**), we observe that simply increasing N_{batch} has a dramatic impact on oracle call efficiency, resulting in up to 537±67 scaffolds for the largest batch size of 2048, a 3.83 fold increase over the RL-AL baseline case and 15.34 fold improvement over pure RL (**Figure 3E**). Importantly, runs with larger N_{batch} also become productive much earlier, requiring only approximately 5000 oracle calls to become productive. The diversity of generated hit scaffolds at the largest batch size (2048) is slightly reduced to 0.252±0.032 compared with 0.199±0.016 and 0.178±0.028 for RL-AL- and RL-baseline respectively (**Figure 3F**), but the significantly larger number of generated hits likely offsets this in practice. Reducing N_{acquired} has the same effect as increasing N_{batch}, resulting in faster lift-off and up to 582±253 scaffolds identified for the smallest N_{acquired} of 32 (**Figure 3G**). The resulting hit scaffold diversity for this case is 0.255±0.022 (**Figure 3H**). The similar behavior of these extreme cases can be rationalized by their similar, low AL/RL ratios – 0.0625 and 0.125 respectively. Variation of N_{loops} at a fixed AL/RL ratio has a much more modest impact, with a larger number of loops leading to modest improvements in terms of hit scaffolds found (167±7 for 8 cycles vs 140±13 for 1 cycle) (**Figure 3C**) with unchanged hit diversity (0.19±0.003 vs 0.208±0.019) (**Figure 3D**)

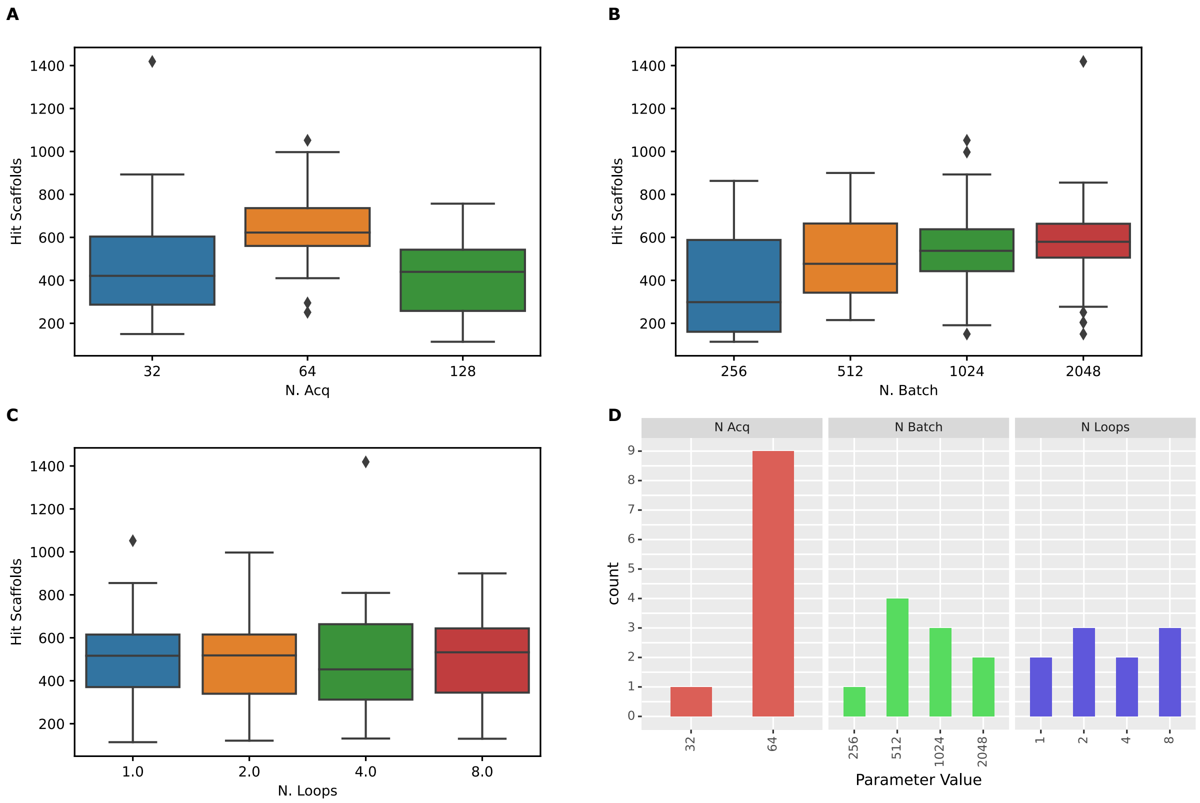

Figure 4: (A-C) The distribution of the number of hit scaffolds identified by a grid-search of 48 RL-AL configurations over 30 000 calls to an ROCs oracle, grouped by N_{acquired}, N_{batch} and N_{loops}. Each boxplot shows the interquartile range of the data and the median as a horizontal line, while the whiskers show minimum and maximum, and outliers are indicated with diamonds. (D) The parameter distribution for the top ten most productive experiments.

Following systematic experimentation of the RL-AL parameters N_{acquired}, N_{batch}, and N_{loops}, it was determined that a synergistic benefit is not achieved by simultaneously decreasing N_{acquired} to its lowest extreme (32) and increasing N_{batch} to its highest extreme (2048). While individually increasing both parameters enhances hit efficiency, N_{acquired} of 64 averages the greatest yield when considering all conditions, outperforming values of 32 or 128 after 30 000 oracle calls (**Figure 4**). The relative diversity of each condition is largely unaffected by the selected settings. The highest average Tanimoto similarity was recorded as 0.277±0.026, with the lowest at 0.23±0.018. Detailed results for each condition can be found in **Supporting Table 2.**

Drawing from these experimental findings, we suggest an optimized RL-AL configuration: Zero weight updates for surrogate-predicted compounds, N_{loops} set to 2, N_{acquired} set to 64, and N_{batch} adjusted to 512, resulting in an AL/RL ratio of 0.125.

Impact of oracle choice

To develop robust methods that work for different oracles, we tested our optimized configuration against both oracle functions, ROCS and ADV (**Figure 5**). Overall, RL-AL optimized drastically improves the oracle call efficiency relative to the RL-baseline, from 35±6 and 818±43 (23 fold increase) to 37±8 and 2 459±273 (75 fold increase) hit scaffolds found over 30 000 oracle calls, for ROCs and docking respectively (**Figure 5A/C**). The solution diversity is slightly reduced, 0.251±0.018 and 0.286±0.005 for RL-AL compared to 0.178±0.003 and 0.162±0.002 for the baseline, which is more than compensated by the much higher number of hit scaffolds found.

In addition to greater numbers of hit scaffolds found over 30 000 oracle calls, the number of oracle calls before the agent becomes productive (i.e., wasted calls) is greatly reduced, with the docking RL-AL system producing 10 hits after only 2 983 ± 247 oracle calls compared to 13 532 ± 4459 oracle calls for the baseline RL case.

However, inspecting the evolution of the MPO score of the generated compounds reveals differences in the oracle behavior that are not present in the RL-only baseline (**Figure 5B/D**). Without RL-AL, the MPO score for runs with both oracles increases steadily as a function of oracle calls to a final value between 0.5 and 0.75. In the case of the ROCs oracle, the MPO score increases rapidly and levels off near 0.9 after ~ 10 000 oracle calls. However, in the case of RL-AL with ADV, the MPO score rapidly increases to ~0.7 and levels off, comparable to the baseline RL state. Investigating this difference, we determined that, while the docking score was rapidly optimized there was a commensurate loss of other components, such as QED, that prevented effective improvement of the MPO score. We hypothesize that this is due to conflicts in the scoring function, for example the addition of hydrogen bond donors might improve docking sores by providing more interactions with the target but push the compound out of the suitable range defined in the QED component.

While this is expected, the goal of the RL process is to improve the MPO score, rather than any specific component. In the case of the zero-weights applied to surrogate predictions, the AL component is effectively selecting which compounds to use in RL likelihood updates. Recognizing this interplay, we propose a final alternative strategy that interprets the MPO score as a probabilistic function of random-valued scoring components and uses this score for acquisition.

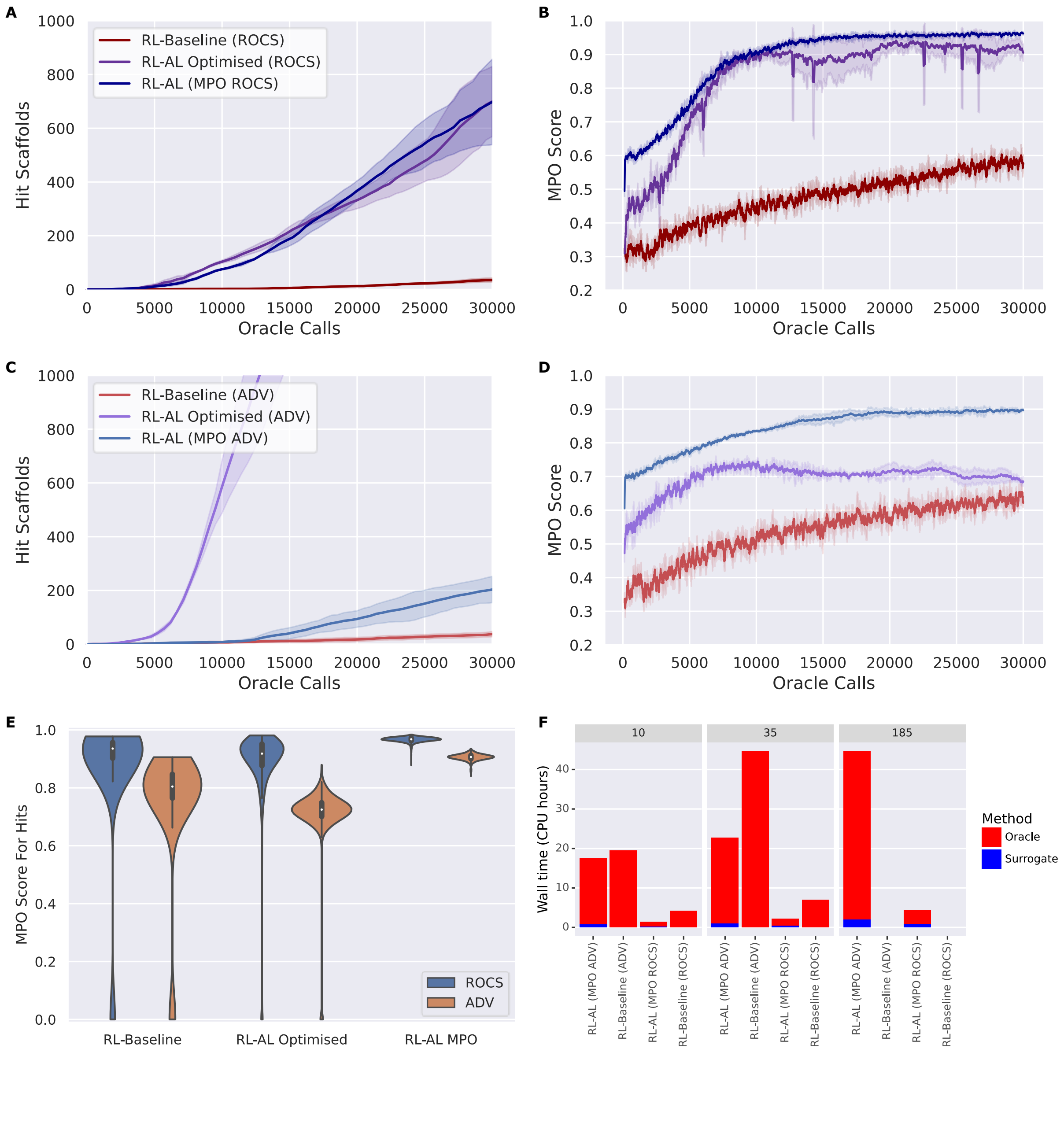

Figure 5: (A-D) The number of hit scaffolds (left) obtained from the optimized RL-AL system and the average MPO of the identified hits (right) as a function of oracle call for two different oracles: ROCS (top) and ADV (bottom), comparing baseline RL, optimized RL-AL, and RL-AL with MPO acquisition. Lines show the mean of three repeats of each experiment while the shading indicates one standard deviation (left). (E) Violin plot of the MPO scores of hit scaffolds obtained from different RL methods and oracles. (F) Bar plot comparing the CPU time needed to sample 10, 35 and 185 hits with different RL methods and oracles, broken down into time spent on the oracle and surrogate modelling process if applicable. The RL-baseline method only identifies 35 hit scaffolds in 30 000 oracle calls, so no times are provided for the other target numbers.

Probabilistic multiparameter active learning

To overcome limitations resulting from over-emphasizing the surrogate-modelled component, we instead propose using the MPO score for acquisition. Instead of performing AL acquisition with respect to the predicted values of the oracle, we acquire based on the predicted distribution of MPO scores, thereby incorporating the other scoring. For example, using a greedy acquisition function we would select compounds to acquire that maximize the expected MPO score, instead of the oracle score.

For common acquisition functions (UCB, expected improvement[15] etc.) we require access to both the expectation and variance of the quantity of interest. This motivates considering the MPO scoring function as a non-deterministic aggregation of scoring components that themselves are random variables. By using the predicted label as a proxy for the oracle scoring function, we generate a distribution of MPO scores by Monte-Carlo sampling (**Methods**).

We explore the effect of MPO acquisition on generated compounds (**Figure 5**). The average MPO score for all compounds generated by the end of the runs are 0.83±0.1 and 0.63±0.13 for the RL-AL optimized and 0.7 ± 0.135 and 0.53 ± 0.27 for the RL Baseline for ROCS and ADV, respectively (**Figure 5B/D**). In the RL-AL MPO case for ROCS and ADV respectively, we increase the average MPO score by ~87% and 56% relative to baseline and ~ 6% and 20% relative to the oracle acquisition RL-AL. To ascertain if the increase in average MPO score results in an increase in the MPO score for hits, we plot the cumulative density of hit MPO scores (**Figure 5E**). We observe that RL-AL optimized produces a higher density of hits at a lower MPO score for ADV, a pattern not observed for ROCS. For both oracles in the RL-AL MPO case all generated hits are found between MPO score [0.85, 1], leading to enrichment of high scoring hits relative to the RL-Baseline and Optimized case where hits are distributed between [0, 1].

To visualize the chemical space coverage of generated compounds we compute a UMAP (Uniform Manifold Approximation and Projection)[69], for all hit scaffolds, using a UMAP model trained on the REINVENT prior (**Supporting Figure 7**). We show that the sampling space is reduced in the RL-AL MPO conditions relative to RL-baseline in the ADV condition, however, we see a compensatory uptick in the sampling of compounds with enriched MPO scores. For ROCS we observe that the MPO strategy both increases the MPO score, and distribution of hits compared to baseline.

For both oracles, the MPO acquisition function leads to acceleration of the hits found per second of computation time. The relative time to find 10 and 35 unique hit scaffolds was significantly reduced by 34% and 32% of the baseline for ROCS and 90% and 50% for ADV. The RL-AL Baseline, in the ROCS and ADV case, finds a maximum of 35 and 37 scaffolds respectively, therefore the time to find more scaffolds cannot be compared directly. In the time it took the RL Baseline to find 35 scaffolds the RL-AL Optimized & RL MPO condition identify ~500 & 185 scaffolds for a ~15 & ~6× enrichment (**Figure 5F**), for ROCS and ADV respectively. For ROCS, selecting either MPO or optimized configurations did not significantly change the number of hits found per second. For RL optimized with ADV, there was a 68- & 11-fold increase in hits found per second compared to the RL-Baseline & RL-AL MPO (Computed from the assumption $\mathrm{AL}_{\mathrm{CPUtime}} - \mathrm{MPO}_{\mathrm{time}} \approx \mathrm{AL}_{\mathrm{CPUtime}}$, see **Supporting Text 1** for hardware information).

Discussion

In this work, we extend the functionality of REINVENT by inclusion of an active learning system for approximating a given oracle function. We demonstrate that this system can be used without a-priori training to iteratively construct a surrogate model and use this model to select subsequent compounds for screening. This RL-AL process consists of two co-dependent loops: the outer RL loop, which generates the design space in each iteration, and the inner AL loop that searches for the best compounds in this space to assess with the oracle. In terms of oracle call efficiency, we report improvements of 14- to 75-fold, depending on the oracle function used.

When optimizing parameters for this process, we identified that the speed of RL optimization is highly sensitive to corruption of the oracle with random noise, which was also reflected in decreased performance when using surrogate-predicted values in place of oracle labels for the RL update. Indeed, we showed increased efficiency by down-weighting surrogate predictions all the way to zero, meaning that we found no benefit from incorporating these predictions into the RL update at all.

Despite not directly using the surrogate model for RL updates, our RL-AL system still demonstrates enormous acceleration in oracle-call efficiency. We believe that the reason for this improvement is related to *curation* of the designs that are screened by the oracle and therefore used for the RL update – the inner AL loop is serving as a filter, using only the most promising ideas to update the RL agent. Since the only mechanism REINVENT-type systems can use to steer molecular generation is to increase the likelihood of generating favorable sequences, it is reasonable that increasing the proportion of positive examples improves convergence, as has been demonstrated in so-called “double loop” reinforcement learning[58] , augmented memory[70], and augmented hill climbing[57]. This intervention did not significantly reduce diversity of the generated leads, possibly due to the relatively permissive nature of the oracle functions.

This complex system depends on several hyperparameters related to the relative size of the RL and AL loops (N_{batch} and N_{acquired}), and our survey of these options demonstrated interdependence between these factors. Generally, the lower the ratio of $\frac{N_{\text{acquired}}}{N_{\text{batch}}}$, the more scope the AL system has to improve upon baseline RL, in rough analogy to the size of virtual library in VS-AL methods[17]. While our optimal configuration provided good performance on both oracle functions tested, it is likely that bespoke optimization of the configuration for the oracle at hand would lead to even better results.

We did observe different behavior in terms of MPO score for the ROCS and ADV oracle, with the ROCS oracle behaving cooperatively with other MPO components. Improving the ROCS score via RL-AL led to commensurate improvements in the MPO score. ADV demonstrated competitive behavior with the other MPO components – anecdotally a common occurrence when using docking-based scoring. In this case, the RL-AL system's acceleration of learning high-scoring components according to the oracle did not lead to any improvements in average MPO score relative to the baseline (although many more “hits” were found).

To address this issue, and incorporate our understanding of the role of the AL loop as a curator of “good examples”, we introduced a novel MPO acquisition function and probabilistic formulation of the REINVENT score aggregation. This system was able to provide massive improvements in MPO optimization, nearly doubling the average MPO score compared with the RL baseline in the ADV case, at the expense of the raw number of hits produced relative to using only the oracle in the acquisition strategy. For the ROCS oracle, the MPO-acquisition strategy performed interchangeably with the oracle-only acquisition strategy.

While simple, our probabilistic reformulation of the REINVENT MPO criteria is highly flexible and we believe it provides a principled way to handle multiple scoring components with varying degrees of uncertainty in RL scoring functions, whether they come from machine learning model confidence (as in this case) or another source (for example, Bennet error[71] in free energy prediction).

Overall, RL-AL provides a self-contained method for accelerating de novo molecular design with RL methods and will provide a substantial reduction in compute resources required to produce the same number and quality of hits. Hopefully this improved sample-efficiency will permit the incorporation of even more accurate and expensive physics-based methods such as alchemical binding free energy predictions, which have shown great promise in VS-based methods[16,17,19,20]. Connecting RL to these methods would enable on-the-fly generation of molecules according to state-of-the-art simulation workflows.

While we have focused on the well-studied application of RL to molecular design, the framework developed here does not explicitly depend on the application setting and offers a promising method to accelerate RL in other domains where oracle experiments or simulations are costly or time-consuming.

Methods and Protocols

Pharmacophore matching with ROCS

Rapid Overlay of Chemical Structures (ROCS [64]) operates on principles of molecular shape and chemical similarity, providing a score for a given query molecule relative to a reference ligand. ROCs is available as a ligand-based scoring function in REINVENT[55]. Both shape similarity and 'color' similarity are evaluated by describing each atom in the target and reference as Gaussian functions and computing the shared volume overlap. For color distributions an additional term describes overlap of chemical groups (donors, acceptors, anions, cations, hydrophobes and rings) are assigned by Implicit Mills Dean forcefield[72]. The Tanimoto Combo Score summarizes the overlap with a score between 0-1. As a target, we selected SC-558, an selective COX2 (Cyclooxygenase-2, **PDB ID: 1CX2**) inhibitor [63]. A ROCs query for SC-558 was prepared using vROCS from the crystallographic pose with the following color features used 2 six membered rings, and one double bonded oxygen, and nitrogen as part of a sulfonamide group. For each query ligand OpenEye's OMEGA[73,74] was used to generate conformers with the following settings: max stereo-centres 0, max conformers 200 and energy window 10 kJ/mol.

Docking with AutoDock Vina

AutoDock Vina (ADV) [66,75] is an open-source program for molecular docking, with a physics-based scoring model for estimating binding affinity of ligands with a protein active site. ADV computation uses van der Waals, electrostatic, directional hydrogen-bond potentials derived from AMBER force field [76], and pairwise additive desolvation term based on partial charges and conformational entropy penalties, to generate the compound score for the protein's pocket of interest. ADV is integrated into REINVENT via the Icolos workflow manager[54]. We selected the Retinoic Acid Receptor alpha (RXRa) as target as ADV has been previously validated to provide good scoring power for this target [66]. The structure of RXRa [65] complexed with inhibitor (S99) was obtained from the Protein Databank (**PDB ID: 7B88**) and prepared according to established ADV protocol[77]. For the generation of three dimensional input structures for ADV w24tilizeded RDKit's embeddings [78].

Virtual Library Screening with Active Learning

We implemented a standard VS-AL approach based on Ref.[17], whereby we trained a random forest (RF) surrogate model using compounds enumerated with physchem descriptors to predict oracle scores and iteratively sampled 128 compounds per AL iteration, selected with the upper confidence bound (UCB) acquisition function. We performed triplicate experiments with different random initial samples. Model parameters were selected based on a retrospective analysis of model error for a property prediction task (**Supporting Text 2**).

REINVENT Generative Model

REINVENT [51,57,79] is an open-source recurrent neural network trained on data derived from ChEMBL [80], which generates tokenized SMILES. REINVENT uses an episodic reward/loss function for policy updates based on the augmented likelihood for a molecule (i.e. token sequence) x:

$$\log P_{\text{aug}}(x) = \log P_{\text{prior}}(x) + \alpha \text{MPO}(x) \qquad (1)$$

Here, $\log P_{\text{prior}}(x)$ is the likelihood of the generated sequence prior model, i.e. the initial state of the agent, which serves a chemical regularizer since the prior is trained to reproduce real molecules from ChEMBL. $S(x)$ is the MPO score assigned to the sequence in (1), computed according to eq. (3), and α is a constant (here, 128) that controls the balance between optimization and retention of the prior. At each epoch, we update the weights of the REINVENT agent to minimize the following modified loss function, termed the weighted difference between augment and prior (wDAP) loss[81], averaged over all molecules x in the batch X:

$$\mathcal{L}(X) = \sum_{x \in X} \frac{w(x)\left(\log P_{\text{aug}}(x) - \log P_{\text{agent}}(x)\right)^2}{\sum_{x \in X} w(x)} \qquad (2)$$

In contrast to previous work, we introduce a weighting function $w(x)$ which assigns a weight between 0 and 1 to all molecules in the batch. This allows us to modulate the contribution of individual compounds to the RL process. We use a learning rate 0.0001 and the Adam Optimiser[82] for RL.

REINVENT includes two non-standard elements that modify the RL process and are used here without modification. Firstly, REINENT uses a memory mechanism called a "diversity filter"[83] to encourage exploration of the chemical space. A record of molecules with scores greater than a user defined minimum is maintained during RL and scores of new molecules in excess of "bucket size" that are too similar to existing ideas are set to zero. Here, we use identical Murcko scaffolds [67], a bucket size 100 and a minimum score 0.2.

Secondly, REINVENT uses an experience replay mechanism whereby a buffer of the top scoring 100 ideas suggestions is maintained and a random sample of 10 SMILES are sampled from this buffer in each RL step and added to the current batch of compounds when computing the loss function and weight updates.

REINVENT Scoring Function

REINVENT conducts multiparameter optimization (MPO) by aggregating over several scoring components $s_i(x)$, computed for each molecule/token sequence x, each converted to floating point values between 0 and 1 with an optional transformation function θ_i, typically a sigmoid function. REINVENT internally maintains a distinction between "penalty", p, components which are always applied multiplicatively and in unweighted fashion, and "non-penalty", np, components with are aggregated with scalar weights between 0 and 1, w_i. Here we consider the geometric mean, although extension to arithmetic mean is straightforward:

$$\mathrm{MPO}(x) = \left(\Pi_{i=1}^{n_p} s_{p,i}(x)\right)\left(\Pi_{i=1}^{n_{np}} \theta_{np,i}\left(s_{np,i}(x)\right)^{\frac{w_i}{\sum_{i=1}^{n_{np}} w_i}}\right) \qquad (3)$$

Note than penalty components return values between 0 and 1 so are not transformed. In this case, “custom alerts” is the only penalty component, and we use $w_i = 1$ for all other components.

Basic REINVENT Configuration

Unless explicitly noted, all REINVENT experiments in used a batch size of 256 compounds per RL epoch, and defined an MPO scoring function according to (3) consisting of the following scoring components:

1) Quantitative estimate of drug-likeness (QED)[84] as a simple metric for drug likeness. QED is a floating point number between 0-1 based on an aggregation of common molecular properties[84]. QED score is computed using RDKit and used without transformation in RL.
2) Molecular weight is used to constrain the size of the generated molecules in the range 200 to 500 Da. The molecular weight for compounds was computed using RDKit and was transformed using a double sigmoid in REINVENT with parameters “Low” = 200, “High” = 550, “Divisor Coefficient” = 550, “Sigmoid Steepness Coefficient” = 20.
3) The number hydrogen bond donors (HBD) is limited to be less than 6 to curtail exploitation of the oracle by undesirable compounds (i.e., adding donors generally improving docking score[53]). The number of hydrogen bond donors (HBD) was calculated using RDKit and transformed using a reverse sigmoid with parameters “Low” = 2, “High” = 6 and “k” = 0.5.
4) A set of “custom alerts” predefined in REINVENT, which prevent generation of unphysical ring sizes and unstable reaction groups (**Supporting Table 3**)

5) The oracle function (docking or ROCS), with an appropriate score transformation to convert the result to range 0 to 1. For use in ROCS, we transform the ROCS score using a sigmoid function in REINVENT with parameters “Low” = 0.3, “High” = 0.65 and “k” = 0.3., chosen such that the reference ligand score 0.6 receives a score of 0.92. Because desirable docking scores are negative numbers (indicating increasing free energy of binding) , we transformed raw docking scores for use in RL using a reverse sigmoid function in REINVENT with parameters “Low” = -13.5, “High” = -6, and “k” = 0.2 . This is chosen so the reference ligand with docking score -11.4 receives a transformed score of 0.73.

Reinforcement Learning with Active Learning

We extend REINVENT’s capabilities by the inclusion of an Active Learning System for approximating expensive oracle functions. Our AL framework is constructed as an external python script which takes as input REINVENT generated compounds as SMILES. Each RL epoch, REINVENT generates N_{batch} compounds via sampling from the current agent state. All inexpensive scoring components (in this case QED, molecular weight, hydrogen bond donors and substructure alerts) are computed for all compounds. Then, an acquisition function is used to select $\frac{N_{\mathrm{aquired}}}{N_{\mathrm{loops}}}$ molecules at a time to be screened by the oracle scoring component (docking or ROCS). Each batch of compounds scored by the oracle is added to the training pool and a surrogate regression model (See **Supporting Text 3** for description of models) is trained to predict the oracle scores for all molecules in the training pool. This process is repeated N_{loops} times per RL step to evaluate exactly N_{aquired} compounds from generated N_{batch} designs with the oracle. Since the acquisition functions depend in general on the surrogate model, we retrain the model on each update. For the first iteration, where there is no data in the training pool, the random acquisition function is used.

The oracle scores (for the acquired compounds) and the predicted scores based on the surrogate model (for un-acquired compounds) are used to compute the MPO score for the compounds in the batch. These scores are used to update the model’s policy in accordance with the original paper (methods, REINVENT generative model).

Construction of appropriate predictive models for molecules is richly studied[85,86], but not the focus of the current work. Based on retrospective testing on a standard RL run, we evaluated several classical (RF, support vector regression, gradient boosting, k-nearest neighbors, Gaussian Process regression) and deep learning approaches (ChemProp) and found limited impact of model choice on surrogate model accuracy (**Supporting Figure 8**). Additionally, we observed limited impact for featurization method (**Supporting Figure 9**). We use RF with RDKit physchem descriptors as a prototypical surrogate model with a 1 000-compound sliding window training set based on the most recently sampled oracle results, with fixed hyperparameters, optimized by retrospective analysis (**Supporting Figure 10**)

Probabilistic Scoring Function

Here, we extend the case in Equation (3) to case where one or more of the scoring components are random variables, i.e., the values of each scoring component are distributed according to some probability distribution $s_i(x) \sim p_i(s|x)$. We indicate realized samples of scoring component i with a second index and no parenthesis $s_{i,j|x}$.We assume that different scoring components are independent, conditioned on the compound in question, which implies that the MPO score is distributed according to a transformed distribution of all of these components $MPO(x) \sim p_{MPO}(MPO|x)$. Since the MPO score is a nonlinear function of the scoring components, both via the geometric mean and the score transforms, it is nontrivial to map nominal uncertainty in scoring components to the MPO score. Therefore, the distribution for scores is estimated via Monte Carlo, that is we compute a expected MPO score for each generated token sequence x:

$$\mathbb{E}[\mathrm{MPO}(x)] \approx \frac{1}{S}\sum_{j=1}^{S}\left(\Pi_{i=1}^{n_p} \mathrm{s}_{p,i,j|x}\right)\left(\Pi_{i=1}^{n_{np}} \theta_{np,i}\left(\mathrm{s}_{np,i,j|x}\right)^{\frac{w_i}{\sum_{i=1}^{n_{np}} w_i}}\right) \tag{4}$$

We find adequate Monte-Carlo convergence with $S = 1000$ samples (**Supporting Figure 11**). An identical approach is used to estimate the standard deviation of the MPO score for use with UCB acquisition.

Training Pool (TRP): The total size (m), and selection of compounds for model training is modulated. Either using chronological ordering, m most recent compounds, or through adaptive subsampling [87]. Adaptive subsampling is a secondary active learning strategy, in which we train our model in iterative stages, by selecting, from the total pool of labelled compounds, those compounds for which the model is most uncertain about in several train, predict, acquire cycles, till m.

Acquisition Strategy: Selection of N_{acquired} is performed through three strategies, Random, Greedy and UCB[88]. The random strategy selects compounds for label acquisition at random and serves as a baseline. With greedy we select the compounds to acquire that optimized the expected oracle score, $\tilde{f}$, over the acquired batch. With UCB, we linearly balance the expected score and compound-specific uncertainty of the surrogate model, $\sigma(x)$, with constant β (here, 1)

$$X_A = argmax_{X_A \subset X;\ |X_A| = N_{\mathrm{aquired}}} \sum_{x \in X_i} \left(\tilde{f}(x) + \beta\sigma(x)\right) \tag{2}$$

Note that in the case of docking we instead minimize (since lower docking scores are better), and that greedy strategy is recovered in the case $\beta = 0$.

Physiochemical Descriptors: We enumerate compound features prior to model training and prediction, and RDKit's implementation of physiochemical descriptors [78]. Physiochemical Descriptors are vectors containing numerical descriptions of compounds physical and chemical properties, such as the number of hydrogen bond donors, lipophilicity, and molecular weight. Features are normalized, using sklearns standard scalar; for sample x the standard score z is computed as follows: $z = \frac{(x-u)}{s}$ where u is the mean of the dataset and s is one standard deviation. Features that are invariant across all compound vectors are removed prior to training and inference.

Chemical Diversity: Compound similarity is measured by computing the pairwise Jaccard coefficient J_{coef} or 'Tanimoto Similarity' of the ECFP (extended connectivity fingerprints) fingerprint (radius 4, 1028 bits) for each compound [89]. The Jaccard coefficient is given by the formula:

$$J_{coef}(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

Whereby the union is the total number of molecular substructures represented in both bit vectors, and the intersection is the number of overlapping bits corresponding to identical substructures in both molecules.

Random Forest Surrogate model: A random forest regressor [90] was implemented using the SciKit-Learn [91]. We retrieve confidence intervals by characterizing the distribution of predictions using its standard deviation. We use hyperparameters max depth = 30, number of estimators = 200, min samples for splitting = 2, based on retrospective analysis with grid search of optimal hyperparameters (**Supporting Text 2, Supporting Figure 2**).

Code availability: The code developed for RL-AL and all inputs and datafiles needed to reproduce the experiments here are provided in the Supporting Information and will be made available on GitHub after peer review.

Acknowledgements

We are grateful to Clara Nyman and Christian Josefson, for their initial research into active learning for reinforcement learning and the Molecular AI department for providing an enriching and collaborative environment. We thank Astrid Boje for careful reading of the manuscript and Megan Pacifici for assistance with graphics.

References

1 Aspuru-Guzik, A., Lindh, R. & Reiher, M. The Matter Simulation (R)evolution. *ACS Central Science* **4**, 144-152 (2018). https://doi.org:10.1021/acscentsci.7b00550

2 Reymond, J.-L. The Chemical Space Project. *Accounts of Chemical Research* **48**, 722-730 (2015). https://doi.org:10.1021/ar500432k

3 Macalino, S. J. Y., Gosu, V., Hong, S. & Choi, S. Role of computer-aided drug design in modern drug discovery. *Archives of Pharmacal Research* **38**, 1686-1701 (2015). https://doi.org:10.1007/s12272-015-0640-5

4 Patel, H. *et al.* SAVI, in silico generation of billions of easily synthesizable compounds through expert-system type rules. *Scientific Data* **7**, 384 (2020). https://doi.org:10.1038/s41597-020-00727-4

5 Gorgulla, C. *et al.* An open-source drug discovery platform enables ultra-large virtual screens. *Nature* **580**, 663-668 (2020). https://doi.org:10.1038/s41586-020-2117-z

6 Acharya, A. *et al.* Supercomputer-Based Ensemble Docking Drug Discovery Pipeline with Application to Covid-19. *Journal of Chemical Information and Modeling* **60**, 5832-5852 (2020). https://doi.org:10.1021/acs.jcim.0c01010

7 Li, J., Fu, A. & Zhang, L. An Overview of Scoring Functions Used for Protein–Ligand Interactions in Molecular Docking. *Interdisciplinary Sciences: Computational Life Sciences* **11**, 320-328 (2019). https://doi.org:10.1007/s12539-019-00327-w

8 Irwin, J. J. & Shoichet, B. K. Docking Screens for Novel Ligands Conferring New Biology. *Journal of Medicinal Chemistry* **59**, 4103-4120 (2016). https://doi.org:10.1021/acs.jmedchem.5b02008

9 Lyu, J. *et al.* Ultra-large library docking for discovering new chemotypes. *Nature* **566**, 224-229 (2019). https://doi.org:10.1038/s41586-019-0917-9

10 Wang, L. *et al.* Accurate and Reliable Prediction of Relative Ligand Binding Potency in Prospective Drug Discovery by Way of a Modern Free-Energy Calculation Protocol and Force Field. *Journal of the American Chemical Society* **137**, 2695-2703 (2015). https://doi.org:10.1021/ja512751q

11 Fratev, F. & Sirimulla, S. An Improved Free Energy Perturbation FEP+ Sampling Protocol for Flexible Ligand-Binding Domains. *Scientific Reports* **9**, 16829 (2019). https://doi.org:10.1038/s41598-019-53133-1

12 Gapsys, V. *et al.* Large scale relative protein ligand binding affinities using non-equilibrium alchemy. *Chemical Science* **11**, 1140-1152 (2020). https://doi.org:10.1039/C9SC03754C

13 Schindler, C. E. M. *et al.* Large-Scale Assessment of Binding Free Energy Calculations in Active Drug Discovery Projects. *Journal of Chemical Information and Modeling* **60**, 5457-5474 (2020). https://doi.org:10.1021/acs.jcim.0c00900

14 Sacks, J., Schiller, S. B. & Welch, W. J. Designs for Computer Experiments. *Technometrics* **31**, 41-47 (1989). https://doi.org:10.1080/00401706.1989.10488474

15 Jones, D. R., Schonlau, M. & Welch, W. J. Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization* **13**, 455-492 (1998). https://doi.org:10.1023/A:1008306431147

16 Yu, J., Li, X. & Zheng, M. Current status of active learning for drug discovery. *Artificial Intelligence in the Life Sciences* **1**, 100023 (2021). https://doi.org:https://doi.org/10.1016/j.ailsci.2021.100023

17 Graff, D. E., Shakhnovich, E. I. & Coley, C. W. Accelerating high-throughput virtual screening through molecular pool-based active learning. *Chemical Science* **12**, 7866-7881 (2021). https://doi.org:10.1039/D0SC06805E

18 Konze, K. D. *et al.* Reaction-Based Enumeration, Active Learning, and Free Energy Calculations To Rapidly Explore Synthetically Tractable Chemical Space and Optimize Potency of Cyclin-Dependent Kinase 2 Inhibitors. *Journal of Chemical Information and Modeling* **59**, 3782-3793 (2019). https://doi.org:10.1021/acs.jcim.9b00367

19 Gusev, F., Gutkin, E., Kurnikova, M. G. & Isayev, O. Active Learning Guided Drug Design Lead Optimization Based on Relative Binding Free Energy Modeling. *Journal of Chemical Information and Modeling* **63**, 583-594 (2023). https://doi.org:10.1021/acs.jcim.2c01052

20 Thompson, J. *et al.* Optimizing active learning for free energy calculations. *Artificial Intelligence in the Life Sciences* **2**, 100050 (2022). https://doi.org:https://doi.org/10.1016/j.ailsci.2022.100050

21 Janet, J. P., Ramesh, S., Duan, C. & Kulik, H. J. Accurate Multiobjective Design in a Space of Millions of Transition Metal Complexes with Neural-Network-Driven Efficient Global Optimization. *ACS Central Science* **6**, 513-524 (2020). https://doi.org:10.1021/acscentsci.0c00026

22 Gubaev, K., Podryabinkin, E. V., Hart, G. L. W. & Shapeev, A. V. Accelerating high-throughput searches for new alloys with active learning of interatomic potentials. *Computational Materials Science* **156**, 148-156 (2019). https://doi.org:https://doi.org/10.1016/j.commatsci.2018.09.031

23 Tran, K. & Ulissi, Z. W. Active learning across intermetallics to guide discovery of electrocatalysts for CO2 reduction and H2 evolution. *Nature Catalysis* **1**, 696-703 (2018). https://doi.org:10.1038/s41929-018-0142-1

24 Patronov, A., Papadopoulos, K. & Engkvist, O. in *Artificial Intelligence in Drug Design* (ed Alexander Heifetz) 153-176 (Springer, 2022).

25 Anstine, D. M. & Isayev, O. Generative Models as an Emerging Paradigm in the Chemical Sciences. *Journal of the American Chemical Society* (2023). https://doi.org:10.1021/jacs.2c13467

26 Janet, J. P., Mervin, L. & Engkvist, O. Artificial intelligence in molecular de novo design: Integration with experiment. *Current Opinion in Structural Biology* **80**, 102575 (2023). https://doi.org:https://doi.org/10.1016/j.sbi.2023.102575

27 Yoshimori, A. *et al.* Design and Synthesis of DDR1 Inhibitors with a Desired Pharmacophore Using Deep Generative Models. *ChemMedChem* **16**, 955-958 (2021). https://doi.org:https://doi.org/10.1002/cmdc.202000786

28 Zhavoronkov, A. *et al.* Deep learning enables rapid identification of potent DDR1 kinase inhibitors. *Nature Biotechnology* **37**, 1038-1040 (2019). https://doi.org:10.1038/s41587-019-0224-x

29 Tan, X. *et al.* Discovery of Pyrazolo[3,4-d]pyridazinone Derivatives as Selective DDR1 Inhibitors via Deep Learning Based Design, Synthesis, and Biological Evaluation. *Journal of Medicinal Chemistry* **65**, 103-119 (2022). https://doi.org:10.1021/acs.jmedchem.1c01205

30 Korshunova, M. *et al.* Generative and reinforcement learning approaches for the automated de novo design of bioactive compounds. *Communications Chemistry* **5**, 129 (2022). https://doi.org:10.1038/s42004-022-00733-0

31 Li, Y. *et al.* Generative deep learning enables the discovery of a potent and selective RIPK1 inhibitor. *Nature Communications* **13**, 6891 (2022). https://doi.org:10.1038/s41467-022-34692-w

32 Ren, F. *et al.* AlphaFold accelerates artificial intelligence powered drug discovery: efficient discovery of a novel CDK20 small molecule inhibitor. *Chemical Science* **14**, 1443-1452 (2023). https://doi.org:10.1039/D2SC05709C

33 Zheng, S. *et al.* Accelerated rational PROTAC design via deep learning and molecular simulations. *Nature Machine Intelligence* **4**, 739-748 (2022). https://doi.org:10.1038/s42256-022-00527-y

34 Bilodeau, C., Jin, W., Jaakkola, T., Barzilay, R. & Jensen, K. F. Generative models for molecular discovery: Recent advances and challenges. *WIREs Computational Molecular Science*, e1608 (2022). https://doi.org:https://doi.org/10.1002/wcms.1608

35 Zhang, J., Mercado, R., Engkvist, O. & Chen, H. Comparative Study of Deep Generative Models on Chemical Space Coverage. *Journal of Chemical Information and Modeling* **61**, 2572-2581 (2021). https://doi.org:10.1021/acs.jcim.0c01328

36 Gómez-Bombarelli, R. *et al.* Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Central Science* **4**, 268-276 (2018). https://doi.org:10.1021/acscentsci.7b00572

37 Lim, J., Ryu, S., Kim, J. W. & Kim, W. Y. Molecular generative model based on conditional variational autoencoder for de novo molecular design. *Journal of Cheminformatics* **10**, 31 (2018). https://doi.org:10.1186/s13321-018-0286-7

38 Dollar, O., Joshi, N., Beck, D. A. C. & Pfaendtner, J. Attention-based generative models for de novo molecular design. *Chemical Science* **12**, 8362-8372 (2021). https://doi.org:10.1039/D1SC01050F

39 De Cao, N. & Kipf, T. MolGAN: An implicit generative model for small molecular graphs. arXiv:1805.11973 (2018).

40 Putin, E. *et al.* Reinforced Adversarial Neural Computer for de Novo Molecular Design. *Journal of Chemical Information and Modeling* **58**, 1194-1204 (2018). https://doi.org:10.1021/acs.jcim.7b00690

41 Olivecrona, M., Blaschke, T., Engkvist, O. & Chen, H. Molecular de-novo design through deep reinforcement learning. *Journal of Cheminformatics* **9**, 48 (2017). https://doi.org:10.1186/s13321-017-0235-x

42 Gupta, A. *et al.* Generative Recurrent Networks for De Novo Drug Design. *Molecular Informatics* **37**, 1700111 (2018). https://doi.org:https://doi.org/10.1002/minf.201700111

43 Segler, M. H. S., Kogej, T., Tyrchan, C. & Waller, M. P. Generating Focused Molecule Libraries for Drug Discovery with Recurrent Neural Networks. *ACS Central Science* **4**, 120-131 (2018). https://doi.org:10.1021/acscentsci.7b00512

44 Mercado, R. *et al.* Graph networks for molecular design. *Machine Learning: Science and Technology* **2**, 025023 (2021). https://doi.org:10.1088/2632-2153/abcf91

45 He, J. *et al.* Molecular optimization by capturing chemist's intuition using deep neural networks. *Journal of Cheminformatics* **13**, 26 (2021). https://doi.org:10.1186/s13321-021-00497-0

46 He, J. *et al.* Transformer-based molecular optimization beyond matched molecular pairs. *Journal of Cheminformatics* **14**, 18 (2022). https://doi.org:10.1186/s13321-022-00599-3

47 Yang, Y. *et al.* SyntaLinker: automatic fragment linking with deep conditional transformer neural networks. *Chemical Science* **11**, 8312-8322 (2020). https://doi.org:10.1039/D0SC03126G

48 Irwin, R., Dimitriadis, S., He, J. & Bjerrum, E. J. Chemformer: a pre-trained transformer for computational chemistry. *Machine Learning: Science and Technology* **3**, 015022 (2022). https://doi.org:10.1088/2632-2153/ac3ffb

49 Igashov, I. *et al.* Equivariant 3D-Conditional Diffusion Models for Molecular Linker Design. *ArXiv* **abs/2210.05274** (2022).

50 Schneuing, A. *et al.* Structure-based Drug Design with Equivariant Diffusion Models. *ArXiv* **abs/2210.13695** (2022).

51 Blaschke, T. *et al.* REINVENT 2.0: An AI Tool for De Novo Drug Design. *Journal of Chemical Information and Modeling* **60**, 5918-5922 (2020). https://doi.org:10.1021/acs.jcim.0c00915

52 Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences* **28**, 31-36 (1988). https://doi.org:10.1021/ci00057a005

53 Guo, J. *et al.* DockStream: a docking wrapper to enhance de novo molecular design. *Journal of Cheminformatics* **13**, 89 (2021). https://doi.org:10.1186/s13321-021-00563-7

54 Moore, J. H. *et al.* Icolos: a workflow manager for structure-based post-processing of de novo generated small molecules. *Bioinformatics* **38**, 4951-4952 (2022). https://doi.org:10.1093/bioinformatics/btac614

55 Papadopoulos, K., Giblin, K. A., Janet, J. P., Patronov, A. & Engkvist, O. De novo design with deep generative models based on 3D similarity scoring. *Bioorganic & Medicinal Chemistry* **44**, 116308 (2021). https://doi.org:10.1016/j.bmc.2021.116308

56 Mokaya, M. *et al.* Testing the limits of SMILES-based de novo molecular generation with curriculum and deep reinforcement learning. *Nature Machine Intelligence* **5**, 386-394 (2023). https://doi.org:10.1038/s42256-023-00636-2

57 Thomas, M., O'Boyle, N. M., Bender, A. & de Graaf, C. Augmented Hill-Climb increases reinforcement learning efficiency for language-based de novo molecule generation. *Journal of Cheminformatics* **14**, 68 (2022). https://doi.org:10.1186/s13321-022-00646-z

58 Bjerrum, E. J., Margreitter, C., Blaschke, T., Kolarova, S. & de Castro, R. L.-R. Faster and more diverse de novo molecular optimization with double-loop reinforcement learning using augmented SMILES. *Journal of Computer-Aided Molecular Design* **37**, 373-394 (2023). https://doi.org:10.1007/s10822-023-00512-6

59 Dulac-Arnold, G. *et al.* Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. *Machine Learning* **110**, 2419-2468 (2021). https://doi.org:10.1007/s10994-021-05961-4

60 Gao, W., Fu, T., Sun, J. & Coley, C. W. J. a. e.-p. Sample Efficiency Matters: A Benchmark for Practical Molecular Optimization. *arXiv preprint arXiv:2206.12411* (2022). https://doi.org:10.48550/arXiv.2206.12411

61 Guo, J. *et al.* Improving de novo molecular design with curriculum learning. *Nature Machine Intelligence* **4**, 555-563 (2022). https://doi.org:10.1038/s42256-022-00494-4

62 Mendez, D. *et al.* ChEMBL: towards direct deposition of bioassay data. *Nucleic Acids Research* **47**, D930-D940 (2019). https://doi.org:10.1093/nar/gky1075

63 Kurumbail, R. G. *et al.* Structural basis for selective inhibition of cyclooxygenase-2 by anti-inflammatory agents. *Nature* **384**, 644-648 (1996). https://doi.org:10.1038/384644a0

64 Hawkins, P. C. D., Skillman, A. G. & Nicholls, A. Comparison of Shape-Matching and Docking as Virtual Screening Tools. *Journal of Medicinal Chemistry* **50**, 74-82 (2007). https://doi.org:10.1021/jm0603365

65 Schierle, S. *et al.* Oxaprozin Analogues as Selective RXR Agonists with Superior Properties and Pharmacokinetics. *Journal of Medicinal Chemistry* **64**, 5123-5136 (2021). https://doi.org:10.1021/acs.jmedchem.1c00235

66 Eberhardt, J., Santos-Martins, D., Tillack, A. F. & Forli, S. AutoDock Vina 1.2.0: New Docking Methods, Expanded Force Field, and Python Bindings. *Journal of Chemical Information and Modeling* **61**, 3891-3898 (2021). https://doi.org:10.1021/acs.jcim.1c00203

67 Bemis, G. W. & Murcko, M. A. The Properties of Known Drugs. 1. Molecular Frameworks. *Journal of Medicinal Chemistry* **39**, 2887-2893 (1996). https://doi.org:10.1021/jm9602928

68 Arús-Pous, J. *et al.* SMILES-based deep generative scaffold decorator for de-novo drug design. *Journal of Cheminformatics* **12**, 38 (2020). https://doi.org:10.1186/s13321-020-00441-8

69 McInnes, L., Healy, J. & Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* (2018).

70 Guo, J. & Schwaller, P. Augmented Memory: Capitalizing on Experience Replay to Accelerate De Novo Molecular Design. *arXiv preprint arXiv:2305.16160* (2023).

71 Bennett, C. H. Efficient estimation of free energy differences from Monte Carlo data. *Journal of Computational Physics* **22**, 245-268 (1976). https://doi.org:https://doi.org/10.1016/0021-9991(76)90078-4

72 Mills, J. E. J. & Dean, P. M. Three-dimensional hydrogen-bond geometry and probability information from a crystal survey. *Journal of Computer-Aided Molecular Design* **10**, 607-622 (1996). https://doi.org:10.1007/BF00134183

73 Hawkins, P. C. D., Skillman, A. G., Warren, G. L., Ellingson, B. A. & Stahl, M. T. Conformer Generation with OMEGA: Algorithm and Validation Using High Quality Structures from the Protein Databank and Cambridge Structural Database. *Journal of Chemical Information and Modeling* **50**, 572-584 (2010). https://doi.org:10.1021/ci100031x

74 Hawkins, P. C. & Nicholls, A. Conformer generation with OMEGA: learning from the data set and the analysis of failures. *J Chem Inf Model* **52**, 2919-2936 (2012). https://doi.org:10.1021/ci300314k

75 Trott, O. & Olson, A. J. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *J Comput Chem* **31**, 455-461 (2010). https://doi.org:10.1002/jcc.21334

76 Huey, R., Morris, G. M., Olson, A. J. & Goodsell, D. S. A semiempirical free energy force field with charge-based desolvation. *J Comput Chem* **28**, 1145-1152 (2007). https://doi.org:10.1002/jcc.20634

77 Forli, S. *et al.* Computational protein–ligand docking and virtual drug screening with the AutoDock suite. *Nature Protocols* **11**, 905-919 (2016). https://doi.org:10.1038/nprot.2016.051

78 Landrum, G. RDKit: A software suite for cheminformatics, computational chemistry, and predictive modeling. *Greg Landrum* **8** (2013).

79 Arús-Pous, J. *et al.* Randomized SMILES strings improve the quality of molecular generative models. *Journal of Cheminformatics* **11**, 71 (2019). https://doi.org:10.1186/s13321-019-0393-0

80 Gaulton, A. *et al.* The ChEMBL database in 2017. *Nucleic Acids Res* **45**, D945-d954 (2017). https://doi.org:10.1093/nar/gkw1074

81 Fialková, V. *et al.* LibINVENT: Reaction-based Generative Scaffold Decoration for in Silico Library Design. *Journal of Chemical Information and Modeling* **62**, 2046-2063 (2021). https://doi.org:10.1021/acs.jcim.1C00469

82 Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

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

84 Bickerton, G. R., Paolini, G. V., Besnard, J., Muresan, S. & Hopkins, A. L. Quantifying the chemical beauty of drugs. *Nature Chemistry* **4**, 90-98 (2012). https://doi.org:10.1038/nchem.1243

85 Akiba, T., Sano, S., Yanase, T., Ohta, T. & Koyama, M. in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* 2623–2631 (Association for Computing Machinery, Anchorage, AK, USA, 2019).

86 He, X., Zhao, K. & Chu, X. AutoML: A survey of the state-of-the-art. *Knowledge-Based Systems* **212**, 106622 (2021). https://doi.org:10.1016/j.knosys.2020.106622

87 Wen, Y., Li, Z., Xiang, Y. & Reker, D. Improving Molecular Machine Learning Through Adaptive Subsampling with Active Learning. (2023).

88 Auer, P. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research* **3**, 397-422 (2002).

89 Rogers, D. & Hahn, M. Extended-Connectivity Fingerprints. *Journal of Chemical Information and Modeling* **50**, 742-754 (2010). https://doi.org:10.1021/ci100050t

90 Breiman, L. Random Forests. *Machine Learning* **45**, 5-32 (2001). https://doi.org:10.1023/A:1010933404324

91 Pedregosa, F. *et al.* Scikit-learn: Machine learning in Python. *the Journal of machine Learning research* **12**, 2825-2830 (2011).