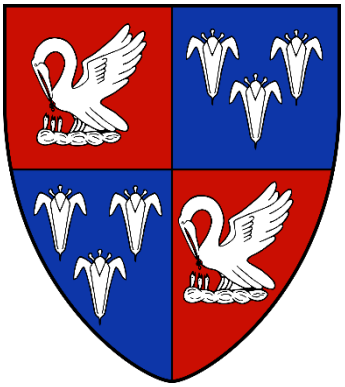
Improving de novo molecule generation for structure-based drug design

Morgan Cole Thomas

Corpus Christi College

University of Cambridge

August 2023



This thesis is submitted for the degree of Doctor of Philosophy

Disclaimer

This thesis is the result of my own work and includes nothing which is the outcome of work done in collaboration except as specified in the text. It is not substantially the same as any work that has already been submitted before for any degree or other qualification except as declared in the preface and specified in the text. It does not exceed the prescribed 60,000-word limit for the Faculty of Physics & Chemistry Degree Committee.

Summary

De novo molecule generation for drug design has seen a resurgence in recent years, mostly due to the rapid advances in machine learning (ML) algorithms that utilise deep neural networks, resulting in a plethora of ML-based generative models. However, there is often a large disparity in published evaluations and applications of such approaches compared to the practical needs of real drug design projects (for example, optimizing QED *versus* optimizing binding affinity commonly approximated by structure-based approaches). Moreover, the density of approaches and often lack of relevant, standardized objectives makes it difficult to truly discern “state-of-the-art”. The work in this thesis aims to address some of these issues and improve the applicability and evaluation of *de novo* molecule generation for practical drug design.

The first research chapter will outline the design and use of an open-source python-based software named MolScore. This configurable suite of scoring functions (including an interface to 5 docking algorithms and ~2,300 trained bioactivity models) can be used to design difficult yet relevant drug design objectives for standardized comparison, or practical usage with generative models. In addition, MolScore includes a graphical user interface to improve usability and a suite of common evaluation metrics to evaluate *de novo* generated molecules.

Next, MolScore was implemented to compare the use of docking as a more difficult objective function for REINVENT (a generative model for goal-directed *de novo* molecule generation), as opposed to more commonly used predictive models of molecule bioactivity. This resulted in increased diversity of *de novo* molecules and improved coverage of known bioactive chemical space. However, the added computational expense required for generative model optimization is a practical disadvantage of docking as a scoring function.

To address the computational expense of optimizing docking scores, a hybrid reinforcement learning algorithm (Augmented Hill-Climb) is proposed to improve the learning efficiency of language-based generative models. This significantly reduced the computational runtime while maintaining the chemical desirability of *de novo* molecules. Augmented Hill-Climb displayed superior efficiency against four other commonly used reinforcement learning algorithms, also displayed in an alternative model architecture. It was then benchmarked against 22 various generative models

showing the best sample efficiency when additionally constraining for chemical desirability.

Overall, the work outlined in this thesis contributes to the field of computational drug design by providing software, algorithmic developments, and benchmark results for different *de novo* molecule generation approaches.

Acknowledgements

Firstly, I thank Sosei Heptares for providing PhD funding and both the University of Cambridge and Sosei Heptares for hosting my PhD studies and all their resources provided, without which, the enclosed work would not have been possible.

I thank my PhD supervisors Andreas Bender and Chris de Graaf for giving me the opportunity to conduct my PhD with them and learn from their valuable expertise. I am incredibly grateful for their academic support in research direction, perspective, networks, writing, reviewing and consistent encouragement to attend and present at conferences, as well as personal support when needed. Equally, I thank Noel O'Boyle for his role and friendship as my unofficial third supervisor who provided all of the above support alongside Andreas and Chris. Furthermore, they have all been my advocates in the research community and within the company of Sosei Heptares for which I will be eternally grateful.

I would like to extend my acknowledgments to many of the Bender group members for their interesting discussions on research topics and comradery (especially important during more challenging times) including, Andrew Boardman, Miguel Garcia Ortegon, Benoit Baillif, Layla Hosseini-Gerami, Anika Liu, Srijit Seal and visitors Koichi Handa, David Figueiredo Vidal and Alexandru Abrudan.

Additionally, I extend my acknowledgments to members of the Sosei Heptares computational chemistry team who embraced my work and offered valuable feedback and expertise on a variety of projects and parallel methodologies including, Rob Smith, Alicia Higueruelo, Pierre Matricon, Brian Bender, Francesca Deflorian, Jonathan Mason, and Sonja Peter.

Last but by no means least, I thank my partner Emily Talbot for all of her understanding of PhD stresses and support through both good and bad times. As well as my family and friends for their consistent support, understanding and patience with everything associated with the PhD candidate package.

List of relevant publications

Some of the content in this thesis is directly taken or derived from previously published work as described below.

Thomas M, Bender A, De Graaf C. Integrating Structure-Based Approaches in Generative Molecular Design. *Curr. Opin. Struct. Biol.* 79, 102559 (2023)
Text, and figures from this work are incorporated in original or adapted from into Chapter 1. This is done under a CC BY 4.0 license.

Thomas M, O'Boyle NM, Bender A, De Graaf C. Augmented Hill-Climb increases reinforcement learning efficiency for language-based de novo molecule generation. *J. Cheminform*. 14, 68 (2022)
Text, figures, and tables from this work are incorporated in original or adapted from into Chapter 4. This is done under a CC BY 4.0 license.

Thomas M, Boardman A, Garcia-Ortegon M, Yang H, de Graaf C, Bender A. Applications of Artificial Intelligence in Drug Design: Opportunities and Challenges. In: Heifetz, A. (eds) Artificial Intelligence in Drug Design. *Methods Mol. Biol*. 2390, 1-59 (2022)
Text, figures, and tables from this work are incorporated in original or adapted from into Chapter 1. Permission was obtained from the publisher.

Thomas M, Smith RT, O'Boyle NM, De Graaf C, Bender A. Comparison of structure- and ligand-based scoring functions for deep generative models: a GPCR case study. *J. Cheminform*. 13, 39 (2021)
Text, figures, and tables from this work are incorporated in original or adapted from into Chapter 3. This is done under a CC BY 4.0 license.

Table of Contents

List of figures

List of tables

Abbreviations

5HT2A / 5-HT$_{2a}$	5-hydroxy-tryptamine or serotonin 2A
ADME	Absorption, distribution, metabolism, and excretion
AGTR1	Angiotensin receptor 1
AHC	Augmented Hill Climb
AI	Artificial intelligence
AnCov	Analogue coverage
AR	Auto regressive
AUC	Area under the curve
AvDS	Average docking score
AZ	AstraZeneca
BAR	Best agent reminder
BBB	Blood-brain barrier
BM	Bemis-Murcko
BO	Bayesian optimization
CASP	Computer-aided synthesis planning
CCDC	Cambridge crystallographic data centre
CDK2	Cyclin-dependent kinase 2
CDK4	Cyclin-dependent kinase 4
CPU	Central processing unit
CS	Cluster size
CSV	Comma separated values
DF	Diversity filter
DMTA	Design-make-test-analyse
DRD2 / D$_2$	Dopamine receptor D$_2$
DS	Docking score
ECFP	Extended connectivity fingerprint
EGFR	Epidermal growth factor receptor
FCD	Fréchet ChemNet distance
FDA	Food and drug administration
FEP	Free energy perturbation
FFNN	Feed-forward neural network
FG	Functional group
GA	Genetic algorithm
GAN	Generative adversarial network
GP	Gaussian process
GPCR	G protein-coupled receptor
GPU	Graphics processing unit
GRU	Gated recurrent unit
GTr	Gated Transformer
GUI	Graphical user interface
HAT	Histone acetyltransferase
HB	Hydrogen bond
HC	Hill Climb
hERG	human Ether-à-go-go-Related Gene
HTS	High-throughput screening
IntDiv	Internal diversity
JSON	JavaScript Object Notation
KL	Kullback-Leibler
LBDD	Ligand-based drug design
LSTM	Long short-term memory
LXR	Liver X receptor
MCF	Medicinal chemistry filters

MCTS	Monte Carlo tree search
MERTK	Tyrosine-protein kinase Mer
ML	Machine learning
MOO	Multi-objective optimization
MPO	Multi-parameter optimization
NLL	Negative log-likelihood
NPR	Normalized principal moments ratio
OPRM1	μ-Opioid receptor 1
OX1R	Orexin receptor 1
PAINS	Pan-Assay Interfering compounds
PDB	Protein data bank
PIDGIN	Prediction including inactivity
PIM1	Proto-oncogene serine/threonine-protein kinase 1
PIS3K	Phosphoinositide 3-kinase
PSO	Particle swarm optimization
QED	Quantitative estimate of drug-likeness
QSAR	Quantitative structure-activity relationship
QSRP	Quantitative structure-property relationship
RA score	Retrosynthetic accessibility score
RF	Random forest
RL	Reinforcement learning
RNN	Recurrent neural network
RS	Ring system
RXR	Retinoid X receptor
SAR	Structure-activity relationship
SAscore	Synthetic accessibility score
SBDD	Structure-based drug design
ScaffDiv	Scaffold diversity/
ScaffUniqueness	Scaffold Uniqueness
SEDiv	Sphere exclusion diversity
SELFIES	Self-referencing embedded strings
SIFTs	Structure interaction fingerprints
SMARTS	SMILES arbitrary target specification
SMILES	Simplified molecular-input line-entry system
SNN	Single nearest neighbour
SVM	Support vector machine
TDC	Therapeutic data commons
TPSA	Topological polar surface area
Tr	Transformer
UMAP	Uniform manifold approximation and projection
US	United States
VAE	Variational auto-encoder
VS	Virtual screening
WGAN	Wasserstein generative adversarial network
XL	Extra large

Chapter 1: Introduction

1.1 Small molecule drug design

Drug discovery is a time consuming and expensive process. It can take 10-15 years[1,2] and up to US$3 billion[1] to discover new therapeutics able to be safely administered to humans for the prevention or treatment of disease. Small molecule drug design is still an integral component of the traditional drug discovery pipeline, as depicted in Figure 1.1. In order for a small molecule to be a suitable drug candidate, it must possess many properties:

On-target activity is required that should translate to clinical outcome, provided a correctly hypothesised target and mechanism of action. At a biophysical level this means a high binding affinity to the target with appropriate binding mode and kinetics. At a pharmacological level this means exerting the right downstream effects such as pathway perturbation. While at a clinical level this means a change in disease endpoint such as reduced tumour growth rates. *In vivo* efficacy is also highly conditional upon pharmacokinetics.

Novelty is required from a legal standpoint to ensure that the molecule doesn't infringe on any existing intellectual property to avoid lawsuits, as well as protect the designer's intellectual property such that the high costs of drug discovery can be recouped. Moreover, any patent outlining intellectual property should be comprehensive enough to make it difficult for rival companies to make trivial changes that can then compete on the market.

Synthetic feasibility is paramount to make and test a molecule. Ideally with a timeframe and difficulty compatible with the design stages. Moreover, a synthetic route must be scalable and sufficiently cost effective such that the compound can be supplied at reasonable quantity and cost and formulation for clinical phases and market distribution, if successful.

Pharmacokinetics determine how the body processes the molecule and is typically broken into absorption, distribution, metabolism, and excretion (ADME) characteristics. Achieving the right balance of these characteristics ensures that a drug can be absorbed in enough quantity, can reach the tissue and target of interest, can be metabolised without adverse effects, and is excreted at an appropriate rate.

Therefore, efficacy is also highly dependent on drug pharmacokinetics such that high enough concentrations of a drug in the right tissue can reach a therapeutic effect, but low enough concentrations such that no adverse events occur.

Selectivity must be achieved such that the molecule only has sufficient activity at the target of interest, and not any undesirable off targets. At best, off-target binding will decrease the free centration of the molecule decreasing availability to bind to the desired target. At worst, off-target binding will result in aberrant pharmacological effects resulting in adverse events or toxicity.

Toxicity must be avoided upon administration of the molecule at a therapeutic dose by either the molecule itself or any of its metabolites. Toxicity can be caused by poor off-target binding, drug-drug interactions, or allergic reactions.

The principal challenge of small molecule drug design is identifying a drug candidate that achieves all of the aforementioned properties to the best degree possible. This is a non-trivial Pareto optimization problem where optimization of one property will lead to a worsening of another. However, this optimization problem is made more complex by assay translation to clinical outcome. For example, biophysical binding affinity is not always predictive of binding affinity in cells, binding affinity is not always predictive of downstream perturbation, downstream perturbation in cells is not always predictive of phenotypic response *in vivo*, and phenotypic response in animal is not always predictive of phenotypic response in human. This iterative removal of assays from true endpoint (i.e., human response) also exists for other properties, such as pharmacokinetics, selectivity, and toxicity. However, this is necessary to keep the cost of discovery down, to better understand mechanism of action and to ethically reduce the number of animals and humans tested.

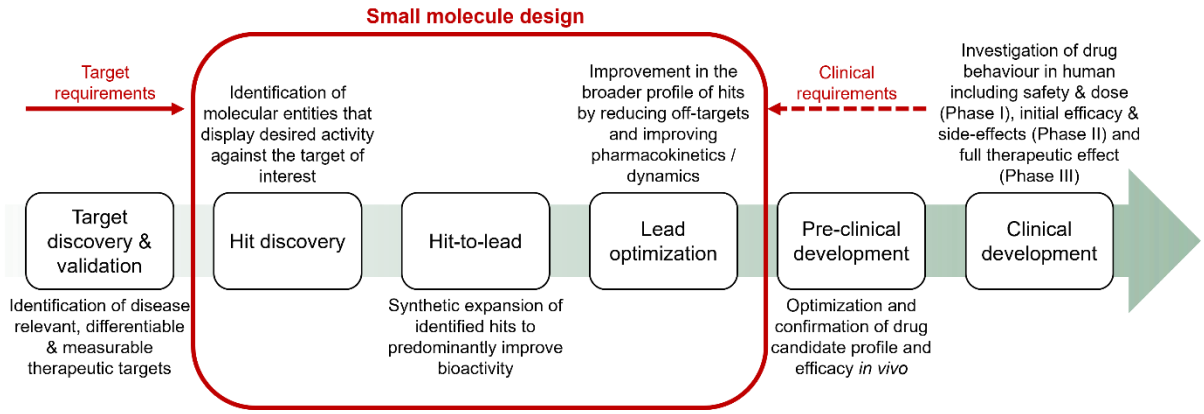


Figure 1.1: Schematic representation of the drug discovery pipeline. The central small molecule design aspects are enclosed within the red box. Other aspects also influence design such as any target requirements (for example, competitive inhibition at an enzyme active site *versus* allosteric modulation) or, pre-clinical/clinical discoveries (for example, poor bioavailability *in vivo* or resistance mutations in the human population) that can return the drug candidate into the design stages.

After target validation, the molecule design process is typically broken into stages by first identifying a 'hit' molecule that possesses some activity–such as on-target binding affinity–in an *in vitro* assay, achieving this usually means that a target ligand is found. Then the chemical space surrounding the hit is explored by synthetic modifications usually to improve binding affinity i.e., improving ligand potency. Lastly, the final congeneric chemical series are identified, and drug candidates are chosen based on the most optimal profile of all properties required, tested on more expensive but predictive *in vivo* assays. This last stage ensures that a ligand has the best chance of translation into a drug.

1.1.1 Hit discovery

To first identify a 'hit' or series of 'hit' molecules, traditional practice is to conduct a high-throughput screen (HTS)[3] where millions of pre-existing library compounds are screened to identify activity in a relevant biological assay, such as protein binding. The caveats of this broad, resource intensive approach are typically low hit rates around 0.5-2%[4,5], the need to filter out frequent hitters[6] or pan-assay interfering compounds (PAINS)[7], and to correct for systematic errors[8]. Moreover, although HTS compound libraries are in the millions of compounds[5], they are still only a small fraction of possible 'drug like' (by Lipinski's definition[9]) chemical space which is likely to be at least 10^{33} compounds (not including stereoisomers)[10]. Therefore, HTS can only evaluate tiny fractions of chemical space that have already been synthesized.

In silico methods can be used to complement or replace HTS, referred to as virtual screening (VS). Such methods include estimation of molecule physicochemical properties, ligand-based similarity methods to compare molecules to a set of reference molecules, or structure-based methods to estimate on-target binding affinity. Any or all of these methods can help select compounds of greater interest in a compound library and avoid compounds predicted to be undesirable. This reduces resource requirement by experimentally screening smaller compound libraries and thus increasing the overall hit rate of HTS[11], for example from 0.02% to 34.8%[12]. An added benefit of VS is the ability to screen and evaluate virtual compound libraries that are

synthesizable on-demand. These libraries increase in size dramatically, for example ENAMINE real space contains approximately 25 billion (based on synthesis success rate of 80%, accessed 10/03/2023), and GSK claims[13] a proprietary library in the order 10^{26}. This, large library still represents a small fraction of estimated chemical space (GSK's "XXL" is 0.00001% of the conservative 10^{33} estimate of drug like chemical space size).

1.1.2 Hit-to-lead and lead optimization

Once hit compounds are identified, the next challenge is to optimize their properties. Hit-to-lead stages generally focus on making structural changes to improve bioactivity, while lead optimization focuses on overall biological, practical, and legal profile as a drug candidate. However, both stages use the same iterative cycle of molecular optimization known as the design-make-test-analyse (DMTA) cycle, shown in Figure 1.2. This four stage cycle is iterated many times during the course of a project, where each cycle can take 4-8 weeks or longer[14,15], be repeated 100s to 1000s of times, and take 1-3 years to progress through hit discovery, hit-to-lead and lead optimization[14,16].

Design corresponds to the proposal of structural changes made to an existing compound that are predicted to improve its properties.

Make corresponds to the synthesis, isolation, and purification of the new set of proposed compounds.

Test corresponds to experimental screening of new compound properties. These are typically *in vitro* assays during hit-to-lead and could be bioactivity assays like on-target binding affinity, physicochemical assays like solubility, or pharmacokinetic properties like cell permeability. Meanwhile, during lead optimization these may include more resource intensive *in vivo* tests to measure e.g., bioavailability.

Analyse corresponds to follow up analysis of the assay results, with a clear link to the 'design' stage both retrospectively (did the previous design result in the expected change) and prospectively (how do the results inform my next design hypothesis).

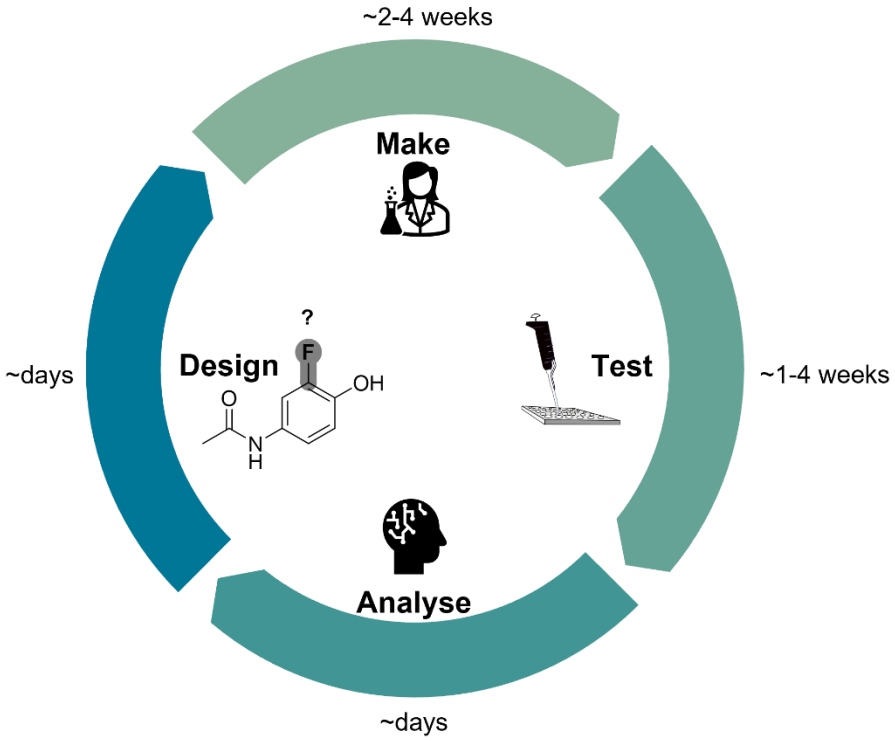


Figure 1.2: Schematic representation of the design-make-test-analyse cycle. Including approximate timeframes of each stage.

In silico methods are routinely used to aid the DMTA cycle, in particular the design, make and analyse stages. Similar methods to VS can be used to aid design such as, ligand-based 2D/3D structural similarity and quantitative structure-activity models (QSAR), structure-based protein-ligand docking simulations, as well as more computationally expensive lower throughput molecular dynamics and free energy perturbation simulations. Computer-aided synthesis planning (CASP) programmes can be used to suggest synthetic routes influencing the 'make' strategy. Finally, data generated from the test stage can not only be used to inform human decision-making but also to update any project-specific predictive models. All of these approaches help to augment human-influenced design and decision-making for more strategic navigation of compound optimization and iteration of the DMTA cycle.

1.2 *De novo* drug design

1.2.1 An in silico design-make-test-analyse cycle

De novo design is an *in silico* concept for the automatic design of new chemical structures. Therefore, *de novo* design can be used in-place of VS for hit discovery or can serve as an *in silico* DMTA cycle that acts as an inner iteration loop within the design stage of an experimental DMTA cycle outer loop (see Figure 1.3). In theory,

moving outer iterations of the DMTA cycle to automated iterations of *de novo* design will help reach ideal solutions, faster and cheaper - reducing the time and resource requirement of hit discovery to lead optimization stages. In practice, recent *de novo* design publications have claimed to achieve the design of a drug candidate in just 21 days[17], a significant acceleration. However, this claim has not come without criticism on scientific impact and chemical novelty[18,19], which is an important reminder that it is drug candidate quality that will make the biggest impact to drug discovery productivity[20]. Overall, due to the potential impact of effective *de novo* design, it is currently an extremely active area of research[21].

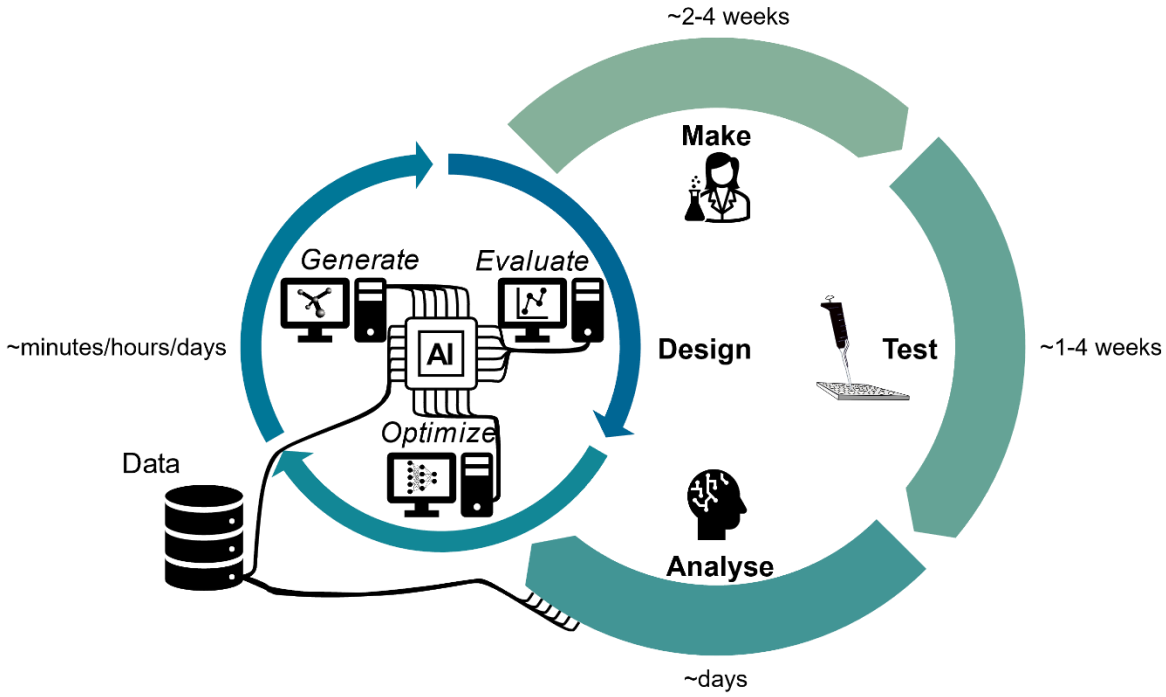


Figure 1.3: Schematic representation of the design-make-test-analyse cycle with *in silico de novo* drug design as an inner loop.

Early *de novo* design algorithms date back to 1989[22], and have been an active area of research for three decades since. As described by Schneider et al.[23], earlier algorithms could be broken down into three constituent parts: (1) how to build/generate a chemical structure, (2) how to evaluate/score molecule quality with respect to a desired endpoint, and (3) how to search/optimize chemical space efficiently. One early example, is the growing of structures by fragments conditional upon a receptor binding pocket, evaluating its steric constraints and hydrogen bonding sites and using a depth or breadth-first algorithm to traverse chemical space, as in Skeletons[22,24]. Many algorithmic variations exist but most comprise of the same three principles; for

example, common search algorithms include depth/breadth-first searches[22,24], random searches[25], Monte Carlo sampling[26] or evolutionary algorithms[27,28]. However, a particular issue with these early, rule-based approaches is inefficient searching and limited sampling of combinatorial explosion associated with the size chemical space.

More recently, there has been a relative resurgence in *de novo* design algorithms from ca. 2016/17 onwards[29,30] with the application of deep generative models (discussed later in 1.2.2.4). Where deep refers to the use of deep neural networks and generative models refers to a class of generative ML models that infer new data instances based on learning a distribution of existing data instances (in probabilistic sense, $P(X'|X)$). In contrast to the previously mentioned rule-based (a.k.a. heuristic) algorithms that must select from a set of atom, fragment, or reaction combination rules to build molecules, deep generative models can use machine learning (ML) to implicitly learn how to build molecules based on inputting datasets of example molecules. The same three principles of build/generate, evaluate/score, and search/optimize remain. The difference is that molecule generation is typically done by sampling from a trained generative model, searching traverses the model's latent space, and optimizing updates the model's parameters. These principles are summarized in Figure 1.4.

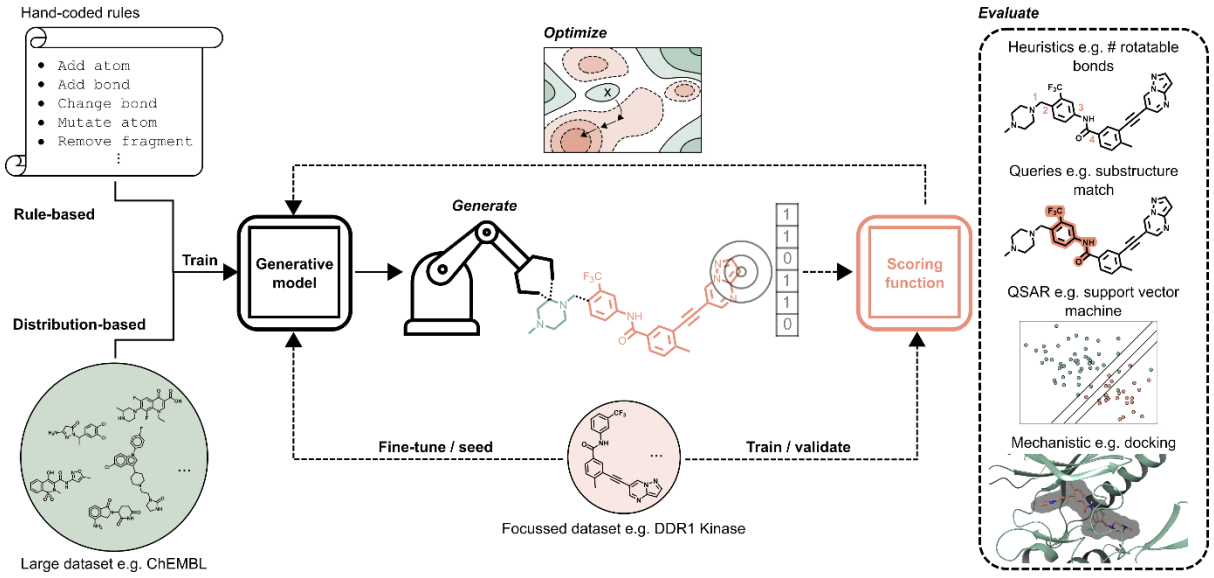


Figure 1.4: Illustration of the generative model principles. Dashed lines represent goal-directed generative models where molecules are generated, evaluated, and optimized in a continuous feedback loop. Molecules can be evaluated by a number of scoring functions.

A key advantage of recent deep generative models for *de novo* design is that they are able to traverse larger chemical spaces far more efficiently than their rule-based counterparts. Rule-based counterparts must have a set of building rules specified, and

when using fragment or reaction rules this already restricts the chemical space to what molecules can be built with the specified rules. Furthermore, chemical space can only be traversed as fast as the minimal combination of rules required to build a molecule. On the other hand, deep generative models have no such restrictions as molecules are generated by simply sampling from the model (e.g., a forward pass of the network) which has learned a probability distribution over chemical space. This learned probability distribution also ensures that *de novo* molecules adhere to similar chemical space properties as input molecules[31], which is not the case using traditional rule-based approaches where chemical space constraints must be applied based on the rules specified. It has further been shown that training a deep generative model on one million compounds (~0.1% of GDB-13[32], a chemical dataset of 975 million molecules) and sampling two billion can recover ~671 million dataset compounds (~70%), where ~78% of sampled *de novo* compounds are in GDB-13 and another ~20% are dataset-like (including repeats)[33]. Based on the model sampling one billion unique dataset or dataset-like molecules, it can be concluded that this model has implicit access to a relevant chemical space ~1000-fold greater than it's input training dataset – which can be further improved by data augmentation strategies[34]. This displays a clear advantage over VS in regard to the chemical space accessible.

For the sake of clarity, I define the following referred to for the rest of this thesis, because terminology is often used interchangeably in the literature. Generative models refer to any *de novo* design algorithm, as opposed to an ML-based generative model in the statistical sense. Rule-based generative models refer to generative models that build molecules based on a set of specified rules. Distribution-based generative models refer to ML-based models that are trained on an input dataset of molecules to implicitly learn molecule building. Goal-directed generative models specify a generative model that is optimized to generate *de novo* molecules that maximize a particular objective. Furthermore, the term 'deep' can apply equally to any generative model, rule- or distribution-based, that integrates the use deep neural networks.

1.2.2 Methods and approaches

The research conducted as part of this thesis uses chemical language models as generative models for *de novo* drug design. Therefore, to embed chemical language

models in a broader context, I will introduce other common approaches but focus in more detail on chemical language models combined with reinforcement learning.

1.2.2.1 Molecular building blocks

For the purposes of *de novo* design algorithms, molecules can be broken down into different types of building blocks and represented in different ways. Figure 1.5 illustrates building blocks and building rules used in *de novo* design to define how to build and generate whole molecules. The main approaches are to build molecules atom-by-atom, by combining fragments, or by combining reactants in a synthetically constrained manner.

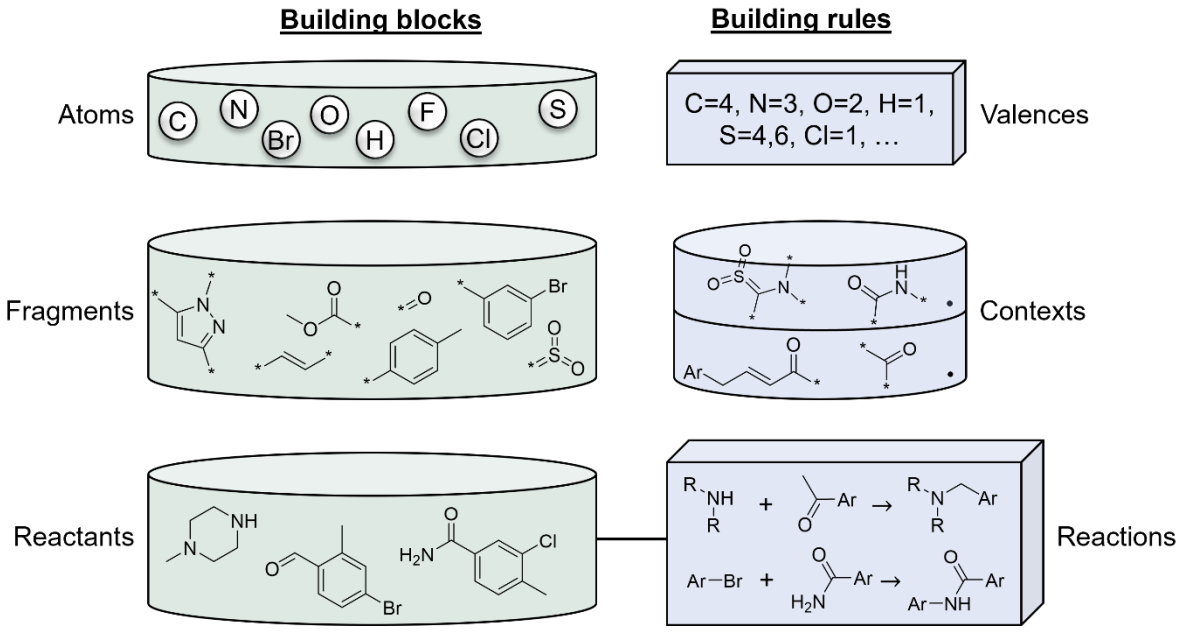


Figure 1.5: Schematic representation of different types of building blocks used in *de novo* design algorithms and building rules that they can adhere to. The simplest being correct valency.

In the most basic sense, atoms can be combined such that a combination doesn't break the rules of chemical valency for joining atoms. This enables full access to chemical space but makes it inefficient to traverse due to combinatorial explosion and can easily result in unstable or non-synthesizable molecules, due to a lack of any other *a priori* chemical knowledge.

On the other hand, fragments can be extracted from databases and combined in new ways resulting in *de novo* molecules that contain known functional groups and substructures. Compared to atom-by-atom combination, less chemical space is accessible, but it is far more efficient to traverse that chemical space. Fragments can also be combined according to valency rules, and synthesizability is more likely, due

to the re-use of fragments existing in known structures. However, this is not guaranteed.

Another approach to improve the likelihood of synthesizability and overall properties of *de novo* molecules is to additionally apply context rules[35]. This additional filter ensures atoms or fragments are only combined when appearing in similar contexts as they appear in chemical databases, where a context may be defined by the nearest surrounding one, two, or three additional atoms. This enforces more *a priori* chemical knowledge during the combination process.

Lastly, sets of reactants and reactions can be used to constrain molecule building to known synthetic reaction resulting in forward synthetic routes as well as *de novo* molecules. This combines reactants with functional groups known to react together according to a set of reaction rules. This guarantees synthesizability of molecules given complete and accurate reaction rules, but even more severely restricts chemical space accessibility. Furthermore, reaction rules are not perfect and can sometimes fail to account for selectivity, competing reactivity or directing groups etc.

These principles, although typically associated with rule-based methods, can equally apply to distribution-based models that, for example, learn a distribution over fragment space[36].

1.2.2.2 Molecular representations

Molecules or their building blocks can furthermore be represented in different ways which then must be encoded such that their representation is machine interpretable for *de novo* design algorithms. Figure 1.6 illustrates some example representations and corresponding machine interpretable encodings, not including 4D representations which are beyond the scope of this work.

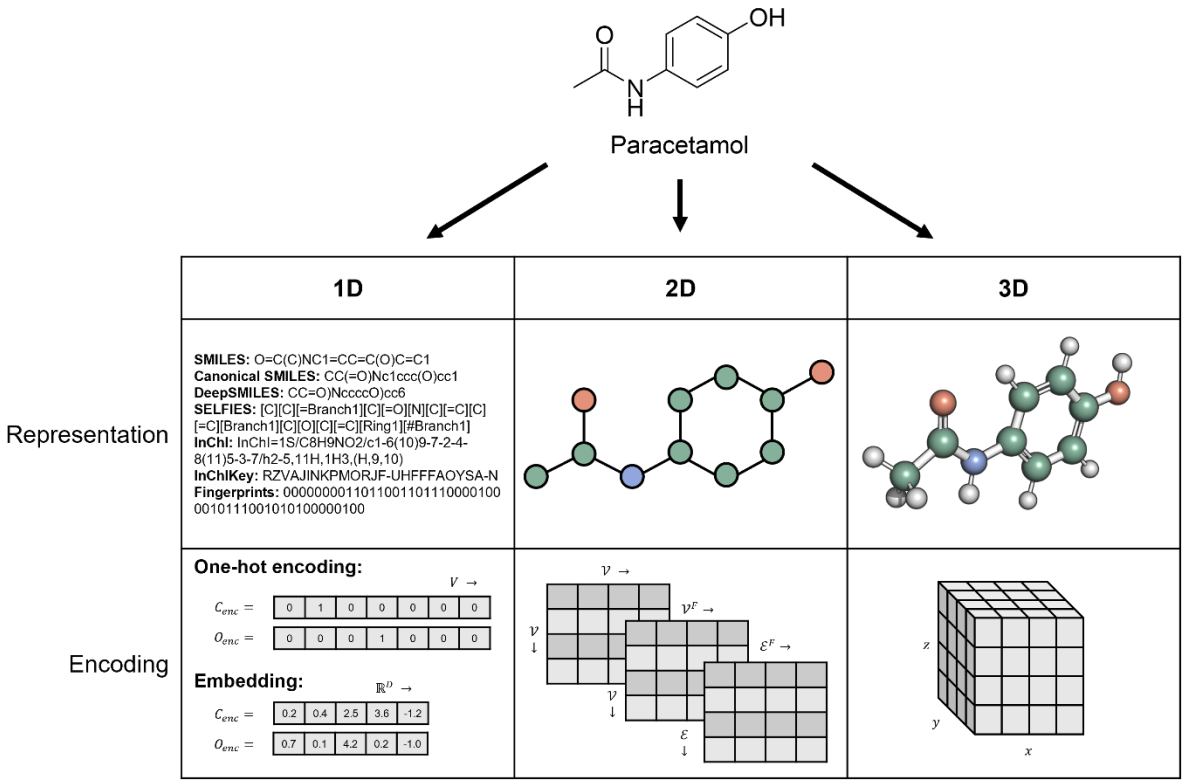


Figure 1.6: Example of different 1D, 2D and 3D representations and encodings to convert the representation into a machine interpretable format.

Although molecules are inherently graph structures, molecular graphs can be also be represented by 1D strings such as SMILES[37] or InChI[38]. However, not all 1D representations are interpretable, unique, or invertible (can be transformed back to a corresponding graph identically), which may be required for different use cases. To combat non-unique representations, canonicalized versions exist that always result in a unique string, which may then depend on a specific canonicalization algorithm used. Moreover, recent adaptions specifically created for use with machine learning have emerged such as DeepSMILES[39], or new grammars altogether such as SELFIES[40]. Not all 1D representations need encoding, for example, non-invertible fingerprints that typically represent the presence of absence of certain atomic environments[41], substructures[42] or other molecular patterns[43,44] are already in binary format ready for machine interpretation. Meanwhile, string-based representations need to be encoded. A classic example is one-hot encoding where an indexed vocabulary is defined based on each unique token t, which could be for example, a letter or word in the string. The encoded representation t_{enc} for a specified token in the vocabulary $t \in V$ is a bit vector with one on bit at the index of the token in the vocabulary, where all other bits are off. This results in a sparse vector representation for each token of dimension equal to the

cardinality of the vocabulary $t_{enc} \in \mathbb{Z}_2^{|V|}$. An alternative, less sparse encoding uses an embedding function to map a vocabulary index into a dense vector of arbitrary dimensions $t_{enc} \in \mathbb{R}^D$. A key advantage of using embedding functions is they are differentiable and learnable, such that similar representations can correspond to tokens with similar semantics, which is not the case for one-hot encoded vectors.

A more intuitive representation of molecules can be defined as an undirected graph $G = (\mathcal{V}, \mathcal{E})$ with vertices $v_i \in \mathcal{V}$ representing atoms, and edges between vertices $e_{i,j} \in \mathcal{E}$ representing bonds between atoms. Moreover, each vertex may have atomic features $v_i^F \in \mathbb{R}$, and each edge bond features $e_{i,j}^F \in \mathbb{R}$. These features can be arranged into matrices as depicted in Figure 1.6, however, the connectivity between nodes still needs to be defined. For this, a symmetric adjacency matrix can be specified indicating which nodes are connected *via* an edge connection. These matrices can then be used as machine interpretable inputs for use deep learning models, such as graph convolution networks[45] or message parsing neural networks[46].

More realistically, molecules exist in a particular conformation i.e., 3D orientation. Therefore, representing them in 3D embeds more information that may be particularly relevant for biological properties such as protein-ligand binding. Moreover, it captures a molecules stereo information inherently, which must be explicitly specified as a feature in 1D and 2D representations to capture. Encoding a molecules 3D representation can be done in several ways, for example, directly converting atomic location in 3D xyz coordinate space to a tensor of specified size and grid spacing (a.k.a. voxel), as illustrated in Figure 1.6. However, care must be taken when encoding 3D information for example, voxels combined with convolution neural networks do not maintain permutation invariance between inputs, such that a translation, rotation, or reflection in input can result in a different function output[47]. Alternatively, the same graph encodings can be used as with 2D representations but, with additional features containing node coordinates, or containing edges between all atoms and respective features containing inter-atomic distances. Whichever representation is used, permutation equivariance and invariance must be considered, for example, specialized graph neural networks that operate on inter-atomic distances[48].

1.2.2.3 Rule-based generative models

Rule-based generative models determine how to combine molecular building blocks according to a set of rules, like those described previously in 1.2.2.1. It should be noted that molecular building blocks can be combined with respective rules either arbitrarily or exhaustively, which is a key approach taken to enumerating virtual compound libraries[49,50]. For *de novo* design however, building blocks and rules are combined in such a way to constrain molecule generation towards a particular objective, hence, the methods discussed here also fit into the category of goal-directed *de novo* design.

Although rule-based *de novo* design explicitly specifies the process of building molecules, it is still necessary to define how to evaluate *de novo* molecules, and how to search chemical space. Early implementations searched chemical space with depth- or breadth-first searches. For example, GenStar[51] which grew molecules atom-by-atom using a depth-first search to continue molecule generation from one top scoring molecule only. However, atom-by-atom depth-first search typically results in shallow regions of local optima due to limited sampling of possible solutions. Other approaches, like RASSE[52], use a breadth-first approach to evaluate more solutions at each stage of atom growing, and conducting atom-by-atom growth on the top 100 solutions at each stage. However, this kind of search is much more computational expensive. Therefore, many early implementations are a combination of depth- and breadth-first search, for example SPROUT[53]. Alternatively, Monte Carlo (i.e., random) search can be used to select atom-by-atom molecular growth, relying on the evaluation method to filter out unfit building blocks and/or rules[25].

An efficient alternative to searching chemical space are evolutionary algorithms, most commonly genetic algorithms. This class of algorithms, inspired by Darwin's theory of natural evolution, operate on populations of molecules that undergo modification to create 'children' that adopt characteristics from their 'parents', are evaluated by a scoring function, undergo selection, and successful 'children' are then added to the next population. A depiction of this iterative procedure is shown in Figure 1.7. Genetic algorithms can operate on multiple molecular representations. Although the building blocks and rules defined in 1.2.2.1 still apply, new operations to create offspring molecules can be defined. Common operations include growth, mutation (replacement of a building block by another) and crossover (linking of two structures together, possibly by a common substructure).

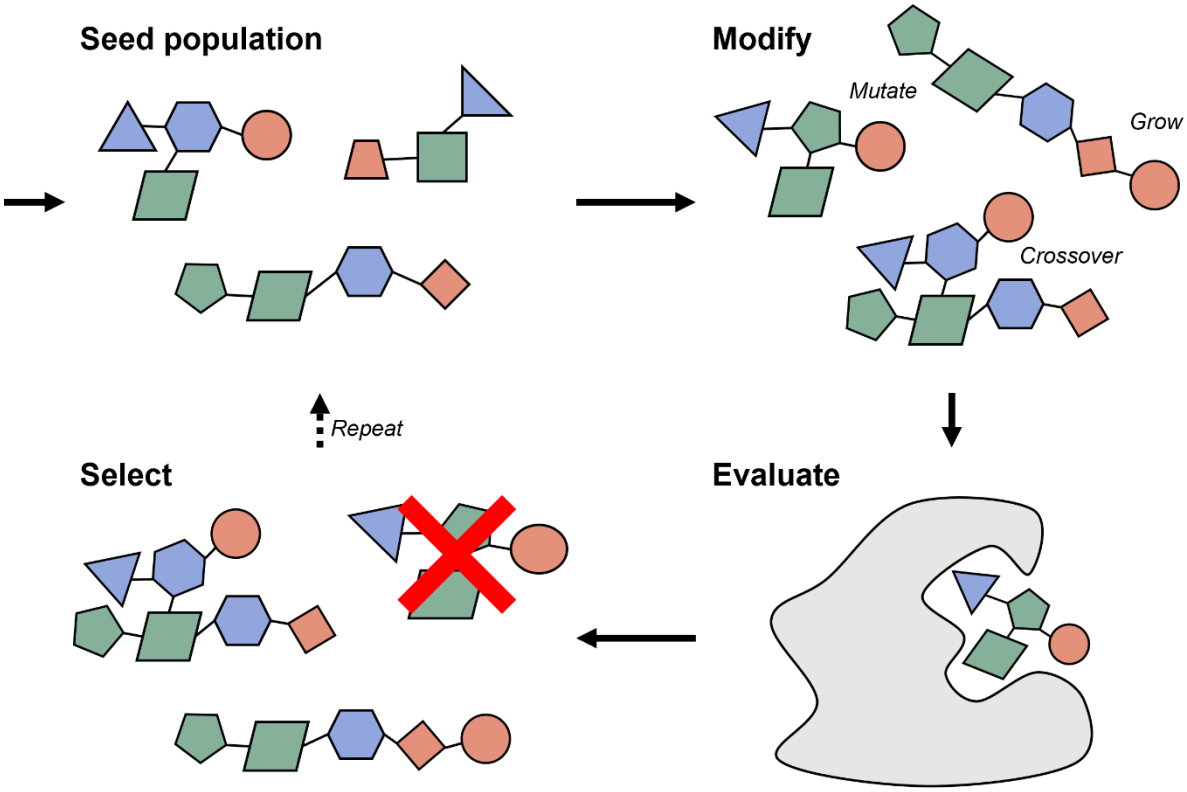


Figure 1.7: Schematic of the iterative cycle conducted by a genetic algorithm for *de novo* design.

Many variations of genetic algorithms exist, which are still top performing generative models[54,55]. Jensen et al.[56] proposed a genetic algorithm operating on graphs using fragment building blocks, which has performed 1st and 2nd on generative model benchmarks[54,55] and shown the ability to optimize difficult scoring functions like docking scores[57]. Operating on 1D SMILES strings provides a greater challenge as mutations to SMILES strings can easily result in invalid SMILES. Nigam et al.[58] circumvented this by operating on 1D SELFIES strings (a recent, representation resulting in 100% valid representations[40]) enabling random mutation operations without the generation of invalid molecules. However, one universal challenge with genetic algorithms is lack of chemical constraint allowing exploitation of non-comprehensive molecule fitness evaluation i.e., iterative optimization can lead to undesirable areas of chemical space and non-drug-like structures. Polishchuk[35] addressed this by enforcing context to genetic algorithm operations, by creating a lookup table of chemical contexts identified in a database of drug like molecules. Thus, avoiding the combination of fragments in contexts which haven't been observed before. Moreover, reaction rules can be followed during modification resulting in molecules much more likely to be synthesizable. Spiegel et al.[59] implemented mutation

operations that followed known reactions resulting in molecule building *via* known synthetic pathways.

Learning how to combine building blocks by building rules can also be interpreted as a reinforcement learning (RL) problem (i.e., learning which actions to take during molecule building). Zhou et al.[60] trained a neural network to predict the value of a particular action (e.g., atom/bond addition/removal) given a state (the current molecule) with respect to maximizing a numerical reward. Then using an ε-greedy approach, the molecule can be built by selecting actions most predicted to provide maximal reward, known as deep Q-networks. Alternatively, the replace of fragments with matched molecular pairs can be learned using actor-critic models to maximize a numerical reward[61]. A similar actor-critic model can also be used to learn which reactants and reaction rules to apply to generate *de novo* molecules[62].

Combinations of genetic algorithms with RL can also be used to increase the efficiency of convergence. For example, by integrating neural networks that select 'parents' in a seed population and which operations to conduct during generation of offspring, trained in an RL setting[63].

1.2.2.4 Distribution-based generative models

The advancement in computational hardware, in particular graphics processing units, have facilitated the growth of deep neural networks (ANNs)[64]. This has led to breakthrough performance for ANNs in a number of domains: image recognition[65], language processing[66], game theory[67], and drug discovery included[68,69]. Although many different types of neural networks exist, Figure 1.8 depicts the most universally used networks. Feed-forward neural networks (FFNNs) directly map an input to an output, recurrent neural networks (RNNs) map an input to an output while sharing information from one state to the next, convolutional neural networks (CNNs) perform convolution filters on higher dimensional grids such as images or voxels, and graph neural networks (GNNs) are a formal framework of FFNNs operating on graph structures. All of which rely on the same principles of (1) processing data through a series of highly parameterized non-linear functions, (2) calculating the error between the network output and expected output, (3) backpropagating the error to identify the gradients of the parameters with respect to the error and (4) modifying the parameters

in the direction associated with decreased error *via* a gradient optimization algorithm such as gradient descent.

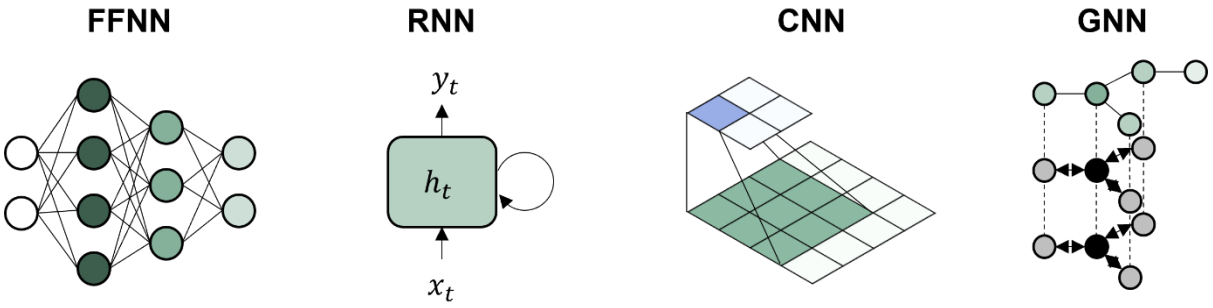


Figure 1.8: Different types of neural networks. From left to right: feed-forward neural networks, recurrent neural networks, convolutional neural networks, and graph neural networks. Reformatted figure based on Chuang et al.[70] under a CC-BY-NC-ND license.

All of the neural networks illustrated in Figure 1.8 have been used in a generative model capacity ($P(x'|x)$) for *de novo* drug design. A high-level illustration of how any (or a combination of) of these networks can be implemented in generative models for *de novo* design is shown in Figure 1.9.

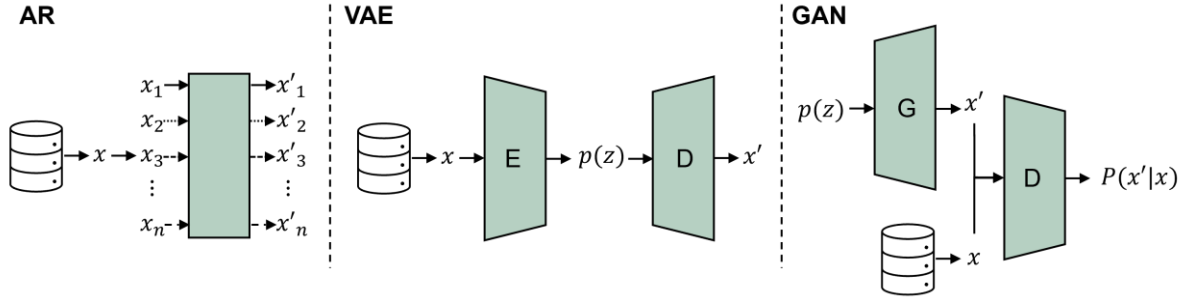


Figure 1.9: Schematic representation of different basic generative model architectures for *de novo* design. From left to right: autoregressive models (for example, recurrent neural networks), variational autoencoders, generative adversarial networks. Autoregressive models predict the next step in molecule building given a partially complete molecule. Variational autoencoders use an encoder (E) to map molecules in a latent embedding, and a decoder (D) to map a latent embedding back to a molecule. Generative adversarial networks use a generator (G) to map a prior distribution (random noise) to a molecule, and a discriminator (D) to classify the predicted molecule as belong to the training dataset distribution or not.

Recurrent neural networks

RNNs are a typical example of an AR generative model. AR models can typically be viewed as sequential models operating in the temporal domain i.e., given past events, predict a future event. RNNs are a special class of neural networks, that not only pass information from input to output, but also from past states to current states (as depicted in Figure 1.10). This introduces the concept of propagating information through time with recurrent forward passes. Therefore, RNNs naturally apply to AR generation due

to the sharing of information through the temporal domain. In order to do this, RNNs have cells (as shown in Figure 1.10) that pass a hidden state h from one state in time t to the next, performing one or more non-linear transformations on the input x, hidden state h and weights w and bias weights b, i.e., neural network operations.

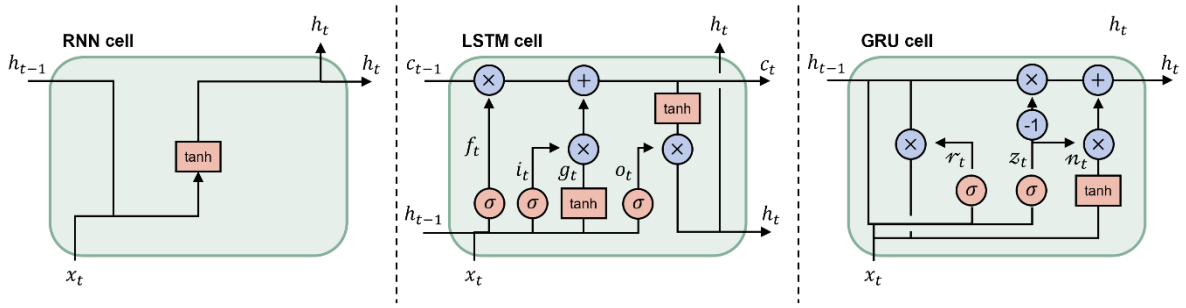


Figure 1.10: Illustration of the three main types of RNN cell used. Left: A vanilla RNN employing only a tanh activation function. Centre: A long-short term memory cell. Right: A gated recurrent unit.

The first implementation was the Elman RNN[71], also commonly referred to simply as RNNs. This RNN cell uses a tanh activation function on the input at a given time step x_t combined with the hidden state from a previous timestep h_{t-1} resulting a single hidden state h_t that is both the cell output and cell state to be used at the next timepoint. This results in the sharing of information between timepoints, however, a key problem faced by this method is exploding or vanishing errors when undergoing backpropagation through time[72].

$$h_t = \tanh(W_{ih}x_t + b_{ih} + W_{hh}h_{t-1} + b_{hh})$$

Equation 1.1

To combat these issues undergoing backpropagation through time, the long short-term memory (LSTM) was proposed by Hochreiter et al.[72]. Firstly, LSTM cells contain two latent vectors shared between timepoints, conventional hidden state h_t as well as a cell memory c_t. This cell has much more explicit control over the flow of information through time, allowing the retention of both long-term and short-term information. The central contributions are commonly recognised as (1) a forget gate f_t that controls the influence of cell memory parameters, (2) input gate i_t and cell gate g_t that controls how much and what influence the input x_t has on the cell memory, and (3) the output gate that controls what contribution the updated cell memory and input have on the output hidden state h_t. These mechanisms protect the cell memory from irrelevant inputs, as well as the output from irrelevant memory. This process is shown in Equation

1.2 where σ refers to the sigmoid activation function and $\odot$ is the Hadamard product (i.e., pairwise multiplication).

$$
\begin{aligned}
f_t &= \sigma\left(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}\right) \\
i_t &= \sigma\left(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}\right) \\
g_t &= \tanh\left(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}\right) \\
o_t &= \sigma\left(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}\right) \\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\
h_t &= o_t \odot \tanh\left(c_t\right)
\end{aligned}
$$

Equation 1.2

More recently, in the interest of compute efficiency, gated recurrent units (GRU) have been proposed in-place of LSTM cells[73]. GRUs contain similar mechanisms to control the flow of information between input, hidden state and output, but require fewer operations and only consist of a single hidden parameter shared between states, eliminating redundancy between h_t and c_t. The central mechanisms are (1) a reset gate r_t that controls how much influence the hidden state has, (2) the update gate z_t that controls the extent of memory maintained, and (3) the new gate n_t that adds new information from the current input. This process is shown in Equation 1.3 below.

$$
\begin{aligned}
r_t &= \sigma(W_{ir}x_t + b_{ir} + W_{hr}h_{t-1} + b_{hr}) \\
z_t &= \sigma(W_{iz}x_t + b_{iz} + W_{hz}h_{t-1} + b_{hz}) \\
n_t &= tanh(W_{in}x_t + b_{in} + r_t \odot (W_{hn}h_{t-1} + b_{hn})) \\
h_t &= (1 - z_t) \odot n_t + z_t \odot h_{t-1}
\end{aligned}
$$

Equation 1.3

The recurrent nature of RNNs lends itself to sequence prediction problems such as natural language, in particular for use in autoregressive sequence prediction as in this thesis. Figure 1.11 depicts a 2-layer deep RNN rolled out through time. At each timestep, a token is input, processed by the RNN, and a probability distribution over all tokens in a vocabulary is output. The final linear layer projects the output into the dimension of the vocabulary, and softmax layer ensures all values sum to one – forming a categorical probability distribution. At the next timestep, the probability distribution is processed by both the current input parameters and hidden parameters taken from the previous state. Once an RNN has been trained, this learned probability distribution can be sampled to predict the next token in a sequence, given previous tokens observed. Each token can be recurrently fed back into the network autoregressively finally resulting in full sequence generation.

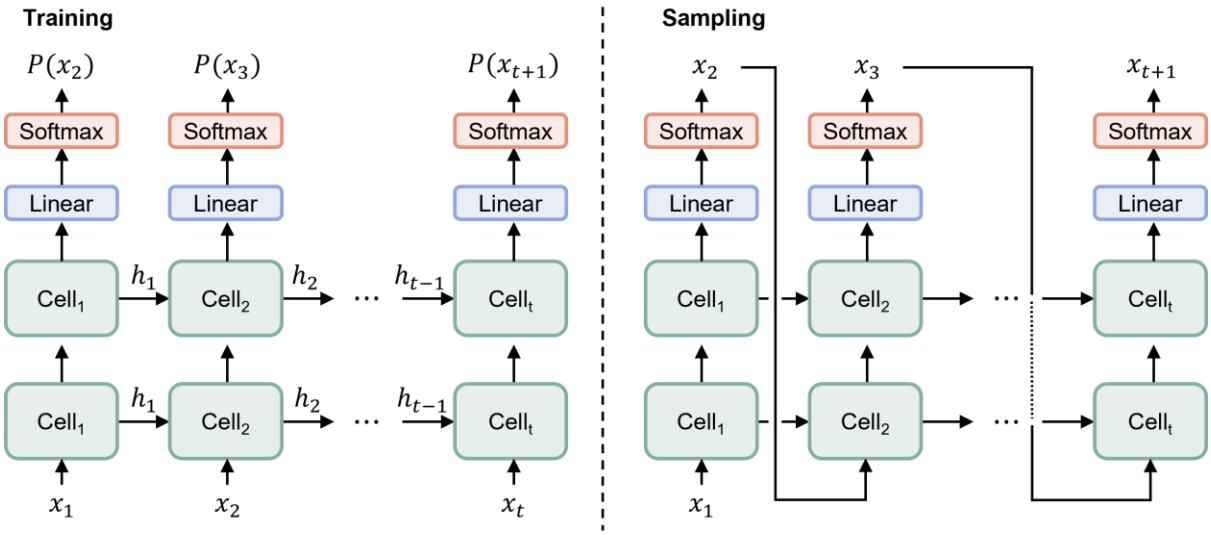


Figure 1.11: Illustration of a 2-layer RNN rolled out through time. At each time step t the network predicts the probability distribution of the next token in a sequence, given the current token. A final layer transforms the output into the dimension of the vocabulary, and softmax layer ensures all logits sum to one to form a probability distribution. Left: the network during training where the correct input token at each timestep is used explicitly (known as teacher forcing[74]), outputting a probability distribution. Right: the network during sample, where the next token is predicted by sampling from the learned probability distribution and then is fed back into the network.

For the neural network to learn the categorical probability distribution over possible tokens, it is trained using maximum likelihood estimation to maximize the likelihood assigned to the correctly predicted token at each timestep. In practice, the equivalent negative log-likelihood (NLL) is used instead such that minimizing this value results in maximizing the likelihood. This avoids hardware precision problems when dealing with very small numbers that may arise from backpropagation through time. Given this can be viewed as a single-class classification problem, this is simply the negative of the log of the probability assigned to the correct token i.e., $P(x_t)$, and is also equivalent to the cross-entropy loss. To compute the NLL across a sequence, the NLL at each timestep is summed (see Equation 1.4). This NLL term can also be used as a representation of how likely a sequence is to be generated by random sampling of the model[33].

$$NLL = -\sum_{t}^{T} logP(x_t|x_{t-1}, \dots, x_0)$$

Equation 1.4

When applied to SMILES-based *de novo* molecule generation, SMILES tokens can be one-hot encoded into a binary vector or embedded into a dense vector and input to the network. As there is no canonical way to start or end a SMILES string, a token for

the start and end of a sequence is inserted. Then, given a large corpus of SMILES, the network can be trained to predict the conditional probability of a SMILES token given previously observed tokens in the string. Once trained, the start token is inserted, and the next token is randomly sampled from the output probability distribution and fed back into the network. This process is repeated until an end token is sampled, resulting in the autoregressive generation of a SMILES string corresponding to a *de novo* molecule.

Segler et al.[29] were one of the first to demonstrate RNN's powerful performance in AR *de novo* molecule generation by training an RNN on 1.4 million molecules from ChEMBL using one-hot encoded SMILES. This model generated *de novo* molecules that well capitulated the physicochemical properties of the ChEMBL molecules yet introduced novel R-group constituent patterns and novel scaffolds. Moreover, by using transfer learning to re-train the RNN on a smaller, more focussed dataset, *de novo* molecules could reconstruct up to 28% of known molecules in a held-out test set. Retrospectively evidencing the potential to discover new *de novo* molecules conditioned towards a particular endpoint. This transfer learning approach is also the most commonly method experimentally validated (see 1.2.3.7). Although over-training during transfer learning can equally lead to decreased novelty, not only from a chemical perspective (sufficiently novel to constitute a new molecule), but also a legal perspective (sufficiently novel to constitute a patentable area of chemical space)[75]. Other works including those by Olivecrona et al.[76] and Popova et al.[77] utilising RNNs trained on SMILES representations have further demonstrated their ability to generate valid and novel *de novo* molecules that contain similar topological patterns as present in the training dataset. In addition, RNNs trained on SMILES have shown excellent enrichment of databases[33], augment-ability by randomization of SMILES inputs to negate unnecessary learning of canonicalization algorithms[34], and ability to condition *de novo* molecule generation (i.e., learn a probability distribution given molecular properties as well as structure alone $P(x'|x,y)$)[78]. RNNs have also consistently performed well relative to other generative models in benchmark studies such as 2nd in GuacaMol[54], 1st in MOSES[79], 1st in smina docking benchmark[80,81] and 1st in sample efficiency[55,82]. This likely contributes to them being the most commonly implemented model in the literature[83]. Due to their performance, they are the generative model used throughout this thesis.

Transformers

In the wider ML field, attention based language models (a.k.a. transformers)[84] have been at the forefront of breakthrough machine learning advances such as BERT[85] and GPT[86,87], however, they are explored to a lesser extent as *de novo* design algorithms[88–90] compared to RNNs, but can also be used for AR sequence generation.

Transformer models were developed by iterative advances in the use of RNNs for sequence to sequence (seq-to-seq) translation. The seq-to-seq problem is more difficult for RNNs, as there is not necessarily a semantic one-to-one mapping when translating between different sequences, let alone handling sequences of different lengths. Therefore, this was tackled by an RNN encoder-decoder approach, inputting the first sequence (seq_1) into the encoder, passing the encoder hidden parameters or memory c to the decoder, and then conducting autoregressive generation of the second sequence (seq_2) with the decoder[91]; however, this led to difficulty retaining relevant information towards the end of the seq_2, by which point the relevant seq_1 information may have been lost. Subsequently, a proposal was made to re-use c at each timestep during seq_2 generation[73], thus helping to prevent forgetting the encoder memory throughout seq_2 generation. A natural progression of this is to instead enable the decoder to learn which tokens in seq_1 are more important for each token during seq_2 generation. This was implemented[92] as a bidirectional RNN encoder (providing conditional information in both directions at a given timestep) where c is passed to each timestep in seq_2 as a weighted sum of the encoder hidden parameters h_t at each timestep $c = \sum_t^T \alpha_t h_t$, with weights α_t. Where α_t are learnable parameters $\alpha_t = softmax(\sigma(h_t, s_{t-1}))$ based on h_t and the hidden parameters at each timestep in the decoder s_{t-1}. This "implements a mechanism of attention in the decoder", whereby the decoder can "pay attention to" certain timesteps in the encoder more than others. This was soon expanded on by researchers at Google by making the network 8 hidden layers deep (where typically they are too difficult to train after 4) by adding residual connections between layers to combat exploding/vanishing gradients[93]. This overall concept then amalgamated into the attention mechanism employed in transformer seq-to-seq models[84] commonly implemented in large language models today[85,86].

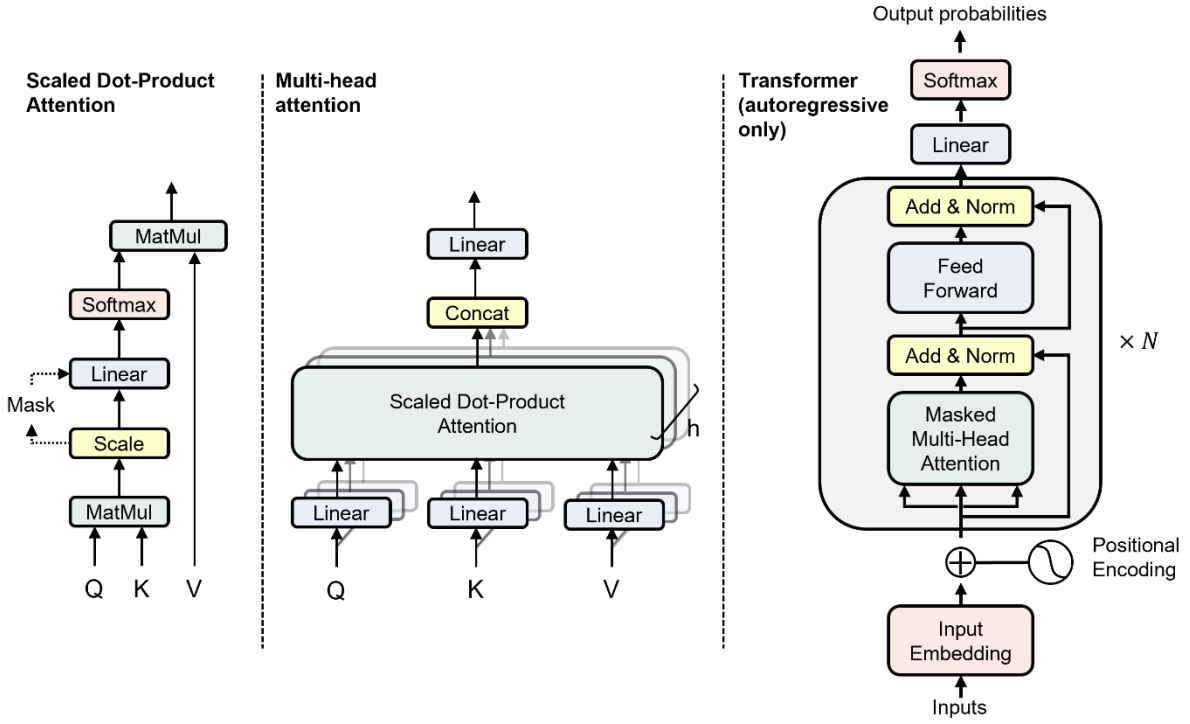


Figure 1.12: Illustration of attention mechanisms in transformer-based autoregressive models based on[84]. Left: Scaled dot-product attention based on queries (Q), keys (K) and values (K), with optional masking. Centre: Linear projection of Q, K and V to form h attention heads. Right: Transformer model for the purpose of autoregression only. Masking is used to force the model to learn sequence prediction autoregressively from left to right and learn the probability distribution over output tokens, as in RNNs.

The attention mechanism shown in Equation 1.5 generalises the calculation of attention between queries (Q) or 'what attention is being applied for', keys (K) or 'what can be paid attention', and values (V) or 'what is being paid attention'. The inputs are analogous to those described in the previous paragraph, where s_{t-1} is the query and h_t is both the keys and values, except that this formalism is applied to the sequence as a whole and so operates on matrices. Queries and keys are combined to calculate the attention weights (analogous to α_t) which then multiply the values (analogous to $\sum_t^T \alpha_t h_t$), as shown in Equation 1.5. In contrast to the attention mechanism described in the previous paragraph, the dot product of Q and K are used to compute attention weights instead of a single layer neural network. This is more compute efficient but must be scaled based on the dimensionality of the keys (d_k) to avoid small gradients in the softmax function[84]. Another difference is optional masking (setting softmax input values to $-\infty$) to avoid attention being paid to them, this can be used to mimic the nature of RNNs that can't observe tokens at future timesteps.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

Equation 1.5

Multi-head attention is a repeating of the attention mechanism to h projected subspaces of the queries, keys, and values, referred to as heads (see Equation 1.6). This allows to jointly attend between information in different subspaces that is inhibited when averaged.

$$MultiHead(Q,K,V) = concat(head_1, head_2, \dots, head_h)W^0$$
$$where\ head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

Equation 1.6

Although designed for seq-to-seq translation with an encoder-decoder architecture, a single encoder can be used for autoregressive *de novo* sequence generation, as shown in Figure 1.12. In contrast to an RNN, a whole sequence is input that is embedded and combined with a positional encoding. This is then processed by N layers, where each layer consists of two sublayers: (1) multi-head attention followed by layer normalization[94] with a residual connection[95], and (2) a fully connected feed-forward neural network followed by layer normalization with a residual connection. Finally, a linear layer followed by softmax is used to project the network values to predict the probability distribution over tokens in a vocabulary. A key difference between this architecture and RNNs is the input of a complete sequence, such that attention can be paid to all tokens in the sequence – making the prediction of the next token trivial by being able to look forward to the next token. To combat this, masking can be used during the attention mechanism to mask any future tokens from being observed. Thus, mimicking the prediction of tokens through time be unmasking the next token iteratively and predicting the probability of the next token at each pseudo-timestep.

This AR transformer architecture, with masking, can be trained identically as to RNNs by minimizing the NLL shown in Equation 1.4. *De novo* sequences can be sampled by identical random sampling of predicted categorical distributions over tokens repeatedly, each time inputting the predicted token and all previously observed tokens. Likewise, it can be applied to SMILES strings identically with the addition of a start and end token. As opposed to RNNs, the attention mechanism should enable better modelling of long-range dependencies contained in SMILES strings (as tokens distant in the SMILES string may correspond to atoms close in the graph).

Application of the transformer architecture for AR *de novo* molecule generation is less common than its RNN predecessor, but it is not unprecedented. Wang et al.[96] reported marginal performance improvement over RNNs according to the MOSES benchmark suite when used for AR generation. Additionally, the authors compared target-specific conditioning of the transformer to target-specific fine-tuning of an RNN resulting in higher predicted activities of transformer-generated *de novo* molecules; with the caveat of different conditioning methods to bias molecule generation for the transformer and RNN confounding any robust interpretation. Moreover, Yang et al.[97] compared an AR transformer with an RNN in a similar fashion and found marginal improvements in SMILES validity from ~95% to ~98% and upon fine-tuning on a target-specific dataset or RL-based optimization, the transformer generated less topologically similar molecules to known target molecules. However, the RNN implemented in this work was unusually shallow, consisting of only one hidden layer with a dimensionality of 1,500 compared to three hidden layers of 512 as seen more commonly[76]. Lastly, Wang et al.[98] found that to achieve the best property-conditioning in combination with RL-based optimization they had to distil the transformer to an RNN before conducting RL, and the result for optimization success was comparable to ordinary RNNs with RL[99]. The benefits of using the more theoretically advanced attention mechanisms in transformers thus appear predominantly in conditional *de novo* molecule generation, however, for unconditional AR generation and especially RL-based optimization there are no clear benefits based on current evidence put forward in their applications. This may explain why the application of transformers is more common in property-conditioned sequence to sequence translation tasks (as per transformers original application) such as, scaffold hopping[100], scaffold decoration[89,101] or substructure transformation[88,102].

Graph-based autoregressive models

AR *de novo* molecule generation is not exclusive to RNNs, as GNNs can also be used. For example, Mercado et al.[103] deconstructed a training dataset into constituent building paths and trained a GNN to reconstruct molecules by predicting action probability distributions that can be sampled during *de novo* molecule generation. However, this didn't particularly outperform RNNs with respect to reconstructing molecules, ring systems and functional groups in a reference dataset, except for those

present in low frequencies. This benefit also comes at the cost of increased computational expense at both training and sampling time[104].

Variational autoencoders

Variational autoencoders (VAEs) embed molecules into a latent space with an encoder $q_\phi(z|x)$, and then map a latent embedding back to a molecule with a decoder $p_\theta(x'|z)$, whilst also constraining the latent space to adhere to a particular distribution (typically a Gaussian distribution). VAE's are trained using maximum likelihood to increase the probability of molecule reconstruction by the decoder, as well as the to minimize the distance between the latent space distribution and a prior distribution using Kullback-Liebler divergence (see Equation 1.7). Once trained, this allows sampling of new latent representations from the learned probability distribution in latent space, that can be decoded into *de novo* molecules. Several variations exist for example adversarial VAEs and supervised VAEs[105].

$$\mathcal{L}_{\theta,\phi} = E_{z\sim q_\phi(z|x)}\left[log p_\theta(x)\right] - KL(q_\phi(z|x), p(z))$$

Equation 1.7

One of, if not the first ML-based generative model was a VAE proposed by Gómez-Bombarelli et al.[30]. Where the authors implemented an RNN encoder and decoder to map SMILES into a latent embedding and back to reconstructed SMILES again. Although the VAE performed similarly to rule-based methods with respect to reconstructing test set molecules, it was able to better constrain the property space to be more similar to the training dataset. Moreover, the distance in latent embedding corresponded to distance in chemical space, and when combined and jointly trained with a FFNN to predict molecular properties from their latent embeddings, the latent space is organized such that molecules with similar properties occupy similar regions of latent space too. However, there were some caveats to this approach, including low validity rates (~70% valid *de novo* molecules with 1,000 attempts per latent embedding), and undesirable non-drug-like substructures.

To address low chemical validity, other implementations have integrated VAEs operating on graphs and an intermediary junction tree encoding-decoding to better ensure the validity of generated molecules, resulting in 100% validity and more stability with respect to idiosyncrasies of *de novo* structures[106]. New 100% validity guaranteed

formal grammars have also been partially inspired by this problem, showing the circumvention of invalid *de novo* molecules in VAEs[40].

Lastly, VAEs can be combined with AR approaches, for example, Maziarz et al.[107] combined an encoder-decoder GNN with AR-style building block prediction: training a decoder to predict molecule building at different steps with a combination of the latent embedding and partially complete molecule. Performing well on benchmarks but crucially enabling scaffold constrained generation trivially.

Generative adversarial networks

At a high level, generative adversarial networks (GANs) can be seen as switching the order of VAEs, where instead of first mapping molecules to a latent distribution $q_\phi(z|x)$ GANs map a fixed prior distribution $p(z)$ to molecules $p_\theta(x'|z)$. Where $p(z)$ is typically a Gaussian distribution of random noise. Learning how to generate molecules from noise is done by adding a discriminator that learns to classify whether the generated molecules belong to the training set distribution or not. Thus, the purpose of the generator is to fool the discriminator, and the purpose of the discriminator is to differentiate the generator, hence the adversarial terminology. This can be interpreted as a min max game as shown in Equation 1.8, where G is the generator and D is the discriminator. In practice, however, this is unstable due to small gradients during the initial stages of training, and so a more stable alternative referred to as Wasserstein GAN (WGAN) is used that frames the problem as minimizing the probability distributions between the generator and real distribution[108].

$$\min_G \max_D (D, G) = \mathbb{E}_{x \in p_d(x)}[logD(x)] + \mathbb{E}_{z \in p_z(z)}[\log(1 - D(G(z)))]$$

Equation 1.8

Guimaraes et al.[109] introduced GANs for molecular *de novo* design by training a WGAN to generate SMILES by using an RNN generator and CNN discriminator. This displayed GANs ability to generate valid SMILES at a rate of above 90%. Moreover, this was combined with RL for molecular property optimization. Many other forms of GANs have since emerged, for example, utilising GNNs[110], conditioned on gene expression profiles[111], or to generate ligand pharmacophores using CNNs with voxels for downstream captioning[112]. However, GANs appear to be less commonly used than other distribution-based generative models[83], possibly due to more complex training model requirements[105].

1.2.2.5 Optimization algorithms for goal-directed design

While it is useful to be able to implicitly learn how to generate *de novo* molecules from data *via* distribution-based generative models, it is ever more desirable to be able to bias molecule generation towards a particular endpoint. In order to achieve this, several optimization algorithms can be used in combination with a scoring function that scores *de novo* molecules in relation to a desirable property. Other methods can be used to bias molecule generation by directly using data available, for example, transfer learning (a.k.a. fine-tuning) by re-training a generative model on a smaller, more relevant dataset[29]; or conditioning by learning a joint probability distribution over training data and their respective properties and then 'steering' *de novo* molecule generation based on input desirable properties[113].

Heuristic algorithms

Heuristic algorithms can be used to search within the latent space learned by a generative model. Yang et al.[114] used this approach by applying a Monte Carlo tree search (MCTS) over an RNN trained on SMILES. This is used to determine the sampling of tokens from the RNN, where the learned probability distribution by the RNN improves the relevance of the rollout procedure. This was shown to outperform the use of Bayesian optimization for the optimization of an arbitrary objective (penalized logP). Another approach taken by Xu et al.[115] was to conduct a simple greedy search in the learned latent space of VAE. This was done by sampling the nearby latent space around a reference compound, assigning the respective decoded molecule with the best evaluated score as the new reference compound, and repeating. This approach managed to improve the docking score ~1.7-fold over the course of sampling ~20,000 molecules.

A caveat of applying heuristic search algorithms to the learned latent space of generative models is the restricted search domain of the generative model. In other words, no generative model parameters are updated resulting in a fixed applicability domain of the generative model based on any initial training, therefore, it can be highly dependent on the breadth of chemical space the model is initially learned.

Evolutionary algorithms

Evolutionary algorithms, in particular GAs are commonly combined with rule-based generative models to optimize towards a particular endpoint (see 1.2.2.3). However,

GAs can also be combined with distribution-based generative models. Ahn et al.[116] used a GA on top of RNN-generated *de novo* molecules to perform genetic operations identifying high-scoring molecules, then fine-tuning the RNN with newly identified high-scoring molecules, all in an iterative loop. This outperformed 16 other benchmarked generative algorithms with respect to optimizing an arbitrary molecular property (penalized logP), including traditional rule-based GAs. Other evolutionary algorithms have also proven effective, such as particle swarm optimization (PSO). This was used by Winter et al.[117] to search the latent space of a trained VAE by a swarm of particles that share knowledge on the potential surface of the search space (evaluated by the scoring function), and congregating in areas of desirable chemical space which can then be decoded to *de novo* molecules. Similar to heuristic algorithms, this search algorithm is constrained by the breadth of learned latent space.

Bayesian optimization

Bayesian optimization is a statistically well-founded approach relying on Bayes theorem to inform the selection of new samples. This relies on a surrogate model to approximate the objective function (i.e., scoring function), and an acquisition function to decide how to sample new molecules. Gómez-Bombarelli et al.[30] used this approach in combination with the property conditioned VAE. A Gaussian process (GP) was trained as a surrogate model to predict the properties of molecules in the latent space, then iterative sampling and re-training of the GP to maximize the predicted property was conducted. This resulted in traversal of the latent space to a more desirable region where more optimal *de novo* molecules could be decoded. Similar Bayesian optimization approaches are commonly combined with VAEs for goal-directed optimization[106,118,119].

Similar to heuristic algorithms, Bayesian optimization conducts a search in a defined chemical space, therefore, the same caveat applies regarding dependence on the learned latent space applicability domain. In addition, Bayesian optimization and more specifically typical surrogate models used (for example, GPs) are known to not scale efficiently to large amounts of data. Moreover, the use of a surrogate model to approximate objective functions with large inaccuracies can result in poor optimization[80], and perform empirically worse than non-surrogate-based alternative optimization[55] algorithms.

Reinforcement learning

Reinforcement learning (RL) introduces algorithms to optimize the sequence of actions taken by an agent to navigate a series of states in an environment to maximize a numerical reward also provided by the environment. More formally, this can be described as how to navigate a Markov Decision Process[120]. This is an episodic task where, given a state $s_t \in S$, an agent must decide an action $a_t \in A$ to take at time step t based on interaction with the environment which informs the agent on the current state it's in and its corresponding reward $r_t \in [0, 1]$. 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 that 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 $R_T = \sum_t^T r_t$, while policy-based RL focusses on identifying the best policy π for selecting actions given a state $\pi(a_t|s_{t-1})$, without necessarily consulting a value function to estimate the absolute value of that state/action.

In the context of chemical language models, the generative model is the agent, a state refers to the current complete or partially complete chemical representation, an action corresponds to selecting the next token, and lastly, a reward for the state by can calculated by an external scoring function. However, when considering many chemical language representations, like SMILES, a partially complete representation does not always correspond to a valid molecular graph. Therefore, a scoring function cannot always be used to assign a reward to r_t to a given state s_t. This complicates the use of value-based RL strategies as state/action values cannot always be calculated. In contrast, policy-based RL strategies do not require a reward for each action/state and as such are more commonly used in this context[54,76,77]. Furthermore, as discussed by Olivecrona *et al*.[76], 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.

A simple policy-based RL strategy is REINFORCE[121], which has previously been used for de novo design[77,122]. This is an 'all-actions' policy-based method because the policy update only requires a sum over all action likelihoods (i.e., the policy) and the return for the whole episode. This can also be interpreted as a scaling of the policy by the return. To update the policy, the agent can be trained to minimize the loss function $\mathcal{L}$

(or cost function) shown in Equation 1.9. Where the agent policy is the same as the NLL seen in Equation 1.4 with the following differences in notation $a_t = x_t$ and $s_{t-1} = x_{t-1}, \dots, x_0$.

$$\mathcal{L}_\theta = \left[-\sum_t^T logP(a_t|s_{t-1}) \right] R_T$$

Equation 1.9

REINVENT[76,99], which is a popular strategy used in the literature, is a REINFORCE type strategy that explicitly regularizes policy updates by adding a prior policy to the loss function. Where the prior policy is the NLL calculated from a fixed copy of the language model (the prior) after training on a corpus of molecules. This regularization ensures that the agent (a copy of the language model that subsequently undergoes policy updates) maintains principles of the policy initially learnt by the prior i.e., how to generate valid *de novo* molecules corresponding to the training distribution. A combination of the prior policy $logP_{prior}(A)$ and return R_t, scaled by scaling coefficient sigma σ, is then used to define an augmented likelihood. This augmented likelihood $logP_{\mathbb{U}}(A)$ intuitively then acts as a 'target policy' for the agent, and the agent policy is updated according to the loss function shown in Equation 1.10, which is now defined as the distance between the agent policy and target policy. For brevity, notation is updated such that $logP(A) = -\sum_t^T logP(a_t|s_{t-1})$.

$$\mathcal{L}_\theta = \left[logP_{\mathbb{U}}(A) - logP_{agent}(A) \right]^2$$
$$where\ logP_{\mathbb{U}}(A) = logP_{prior}(A) + \sigma R_T$$

Equation 1.10

More recently a strategy was proposed that offered modest performance improvement over REINVENT called 'best agent reminder' (BAR)[123]. Although this was implemented on a graph-based generative model, it can be equally applied to a chemical language model using the same principles. This strategy keeps track of the best agent by policy so far, updating it periodically. During policy update, a batch of molecules M (of size 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 1.10 for the respective agents, the weighted average is taken across agents scaled by α, as shown in Equation 1.11. This effectively acts to minimize the agent policy difference to the 'best agent optimal policy' and the 'prior optimal policy', scaled by α.

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

Equation 1.11

Hill-Climb (HC)[122] can also be described as a heuristic optimization method or as an off-policy RL strategy and has been benchmarked by Brown et al.[54] and Huang et al.[124] showing state-of-the-art or near state-of-the-art performance. In this context, HC can also be interpreted as a form of iterative fine-tuning or transfer learning. To update the agent policy, molecules are sampled from the agent and then evaluated to calculate the return for each molecule. The top k ranked molecules by their return are then used to update the agent policy by minimizing their NLL as in Equation 1.4. This updates the policy to increase the likelihood of molecules that result in a higher return.

As some RL strategies, for example REINFORCE and HC, are not explicitly regularized by a prior policy, cost terms can be added during policy updates to enforce a degree of prior policy regularization. This is important in practice to ensure policies found in the training distribution are maintained, but also to not catastrophically forget chemical language syntactics and semantics which will result in invalid structures. A common approach[122,125] is the addition of the Kullback-Leibler (KL) divergence between the prior and agent policies scaled by a scaling coefficient λ, as shown in Equation 1.12.

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

Equation 1.12

RL has equally been applied to VAEs[126] and GANs[110,127] to optimize *de novo* molecule generation. As opposed to heuristic algorithms and Bayesian optimization, RL directly adapts or updates generative model parameters – enabling it to, theoretically, learn new chemical distributions to optimize the objective.

A particular challenge with RL is when the scoring function(s) provides sparse rewards and hence, limited signal for the agent (generative model) to learn from. This can be combatted with many tricks that can be used to improve the performance of RL such as experience replay[128] or inception[99].

1.2.2.6 Integrating structure-based principles into de novo design

Structure-based drug design (SBDD) and ligand-based drug design (LBDD) principles are commonly used by *in silico* computational techniques to aid drug design. However, despite the fact that VS commonly uses SBDD principles when protein structures are available, the vast majority of generative models in the literature only use LBDD to guide *de novo* molecule generation. More and more generative models are starting to integrate SBDD principles typically used in VS to identify small molecules with potentially high on-target binding affinities, a much more relevant drug design challenge that can equally be used in additional contexts like first-in-class.

Why integrate structure-based design?

SBDD concepts, can offer several advantages over LBDD concepts[129]. Retrospectively within a design campaign, the availability of at-least one co-crystallized ligand-protein bound complex enables explicit rationalisation of the structure-activity relationship (SAR) for a particular chemotype, in doing so enabling the identification of favourable protein interactions. Prospectively, this allows rational design changes to maximize protein interactions without relying on knowledge of other ligands, therefore this can more easily lead to novel ligand chemistry or is applicable where ligand data is scarce[130–132]. In contrast, LBDDs inherent reliance on, and bias towards, known chemistry limits its ability to identify new chemistry. This translates to prediction unreliability in novel areas of chemical space when using QSAR models[133]. In combination with generative models, it has been shown that QSAR models can be exploited by optimization of specific model hyperparameters including data split and QSAR model seed[134,135].

On the other hand, SBDD also has its limitations, especially with regards to data availability and conditionality. Structural data can be difficult to acquire, and was historically intractable on larger proteins or membrane proteins such as G protein-coupled receptors[136]. However, the availability of structural data is increasing year on year[137] and technologies such as AlphaFold2[69] and related approaches[138–140] are beginning to provide higher quality structure predictions. This is providing increasing promise for integration of predicted structures into SBDD approaches. Meanwhile, the use of docking as a traditional structure-based virtual screening approach itself is limited by poor correlation with small molecule binding affinity and variable performance on different protein systems[141,142], and so this may be further

exacerbated by input of inaccurately predicted structures. With respect to distribution-based generative models directly learning a distribution over structural data also, the available training data is much more sparse compared to its ligand counterpart. Generative models trained only on ligand data are able to utilize databases[49] of up to 10^{11}, whereas the largest database of experimentally determined protein-ligand complexes has approximately 20,000 entries[143]. Moreover, structural data is conditional upon many more factors, for example, the bound receptor, experimental conditions, as well as the software and modeller responsible for fitting to the electron density. This makes training structure-based distribution models much more challenging. Lastly, a structure is still only a snapshot of a specific protein conformation[144], which is influenced by the co-crystallized ligand (or absence thereof) and may not be the required stabilized conformation to obtain the desired downstream bioactivity. Hence, the effective use of SBDD approaches requires strong domain knowledge and specialist approaches to appropriately manage its limitations.

Approaches to integrate structure-based design

Approaches to integrate protein structure with generative molecular design can be classified into four categories: distribution learning or goal-directed optimization and structure-explicit/implicit, shown below in Figure 1.13.

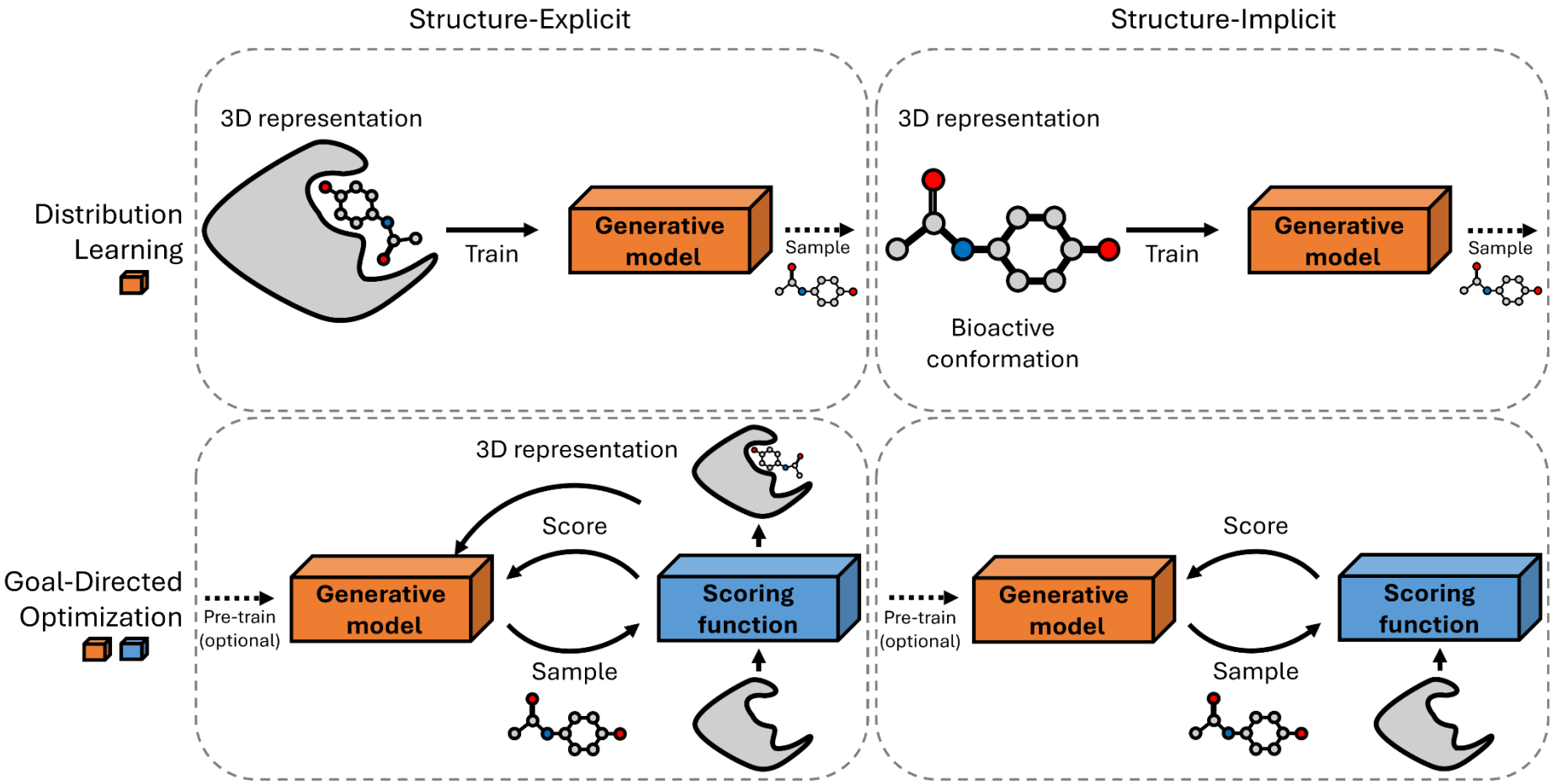


Figure 1.13: Classification of approaches to integrate protein structure into generative molecular design. (Top left) Structure-explicit distribution learning whereby the generative model is trained on representations of the protein (for example, graph, voxel or point cloud) and ligand or bound complex to learn a distribution that can be used for de novo ligand generation. (Top right) Structure-implicit distribution learning whereby the generative model is trained on 3D bioactive conformations of bound ligands that infer some knowledge of protein structure. (Bottom left) Structure-explicit goal-directed optimization whereby a generative model learns to optimize for a structure-explicit scoring function and utilizes a representation of the protein or predicted complex to aid in doing so. (Bottom right) Structure-implicit goal-directed optimization whereby a generative model learns to optimize a structure-explicit scoring function by numerical score alone.

Distribution learning generative models refer to distribution-based, where doing this in a structure-explicit way means that a representation of the protein structure is additionally fed into the generative model, such that it can learn a conditional distribution of small molecules based on protein structure context. Although this could also be done in a structure-implicit way without a protein structure representation (so long as the small molecule is represented in its bioactive conformation which depends on protein structure). Goal-directed optimization goal-directed generative models, where protein structure can be incorporated into the extrinsic scoring function, a key example being the docking score from molecular docking simulations. In the case of docking, I refer to this as structure-implicit if no representation of the protein structure is additionally fed into the generative model. However, a structure-explicit generative model can be achieved by feeding a representation of protein structure (for example, predicted binding pose and surrounding pocket) such that it additionally informs the generative model of future proposals as well as the returned property i.e., docking score. A generative model can theoretically incorporate structure data through both distribution learning and goal-directed optimization approaches which are not mutually exclusive, although this is not seen in practice yet.

Initial structure-based *de novo* design with ML-based generative models in recent years belong to the group of structure-implicit goal-directed optimization, integrating structure information *via* the optimization of structure-explicit scoring functions (namely docking). This avoids the data limitations and complex representation requirement of protein structure information needed for distribution learning. Early work by Cieplinski et al.[80] raised concerns of the potential difficulty of this task with two VAE-based generative models failing to minimize the docking score of *de novo* molecules beyond the top 10% of ZINC[145] (a compound library used for virtual screening). However, this model relied on a surrogate, structure-implicit, model to predict the docking score of molecules given their latent representation (for optimization purposes). Crucially with this approach, surrogate model inaccuracy will compound with the already large inaccuracy observed in the respective docking algorithms, therefore, injecting noise into the scoring. Since this study, this concern has been addressed with the successful optimization of docking scores by a variety of generative models: to evaluate molecule fitness in the case of a genetic algorithm[59], to guide the sampling of latent space as with a variational autoencoder[115,126] or used

to update generative model parameters in a reinforcement learning setting[146,147] (including work as part of this thesis in Chapter 3). These successful approaches all used docking scores directly to inform or update the generative model without reliance on a surrogate model to predict docking scores. However, the use of docking algorithms requires a level of expertise and domain-specific knowledge that some researchers of generative models may not possess. Therefore, easy-to-use tools that are able to conduct docking in a more automated way are extremely valuable. As such, several tools have become available including DockStream[148], TDC[124], DOCKSTRING[149] and MolScore (presented as part of this thesis in Chapter 2)[150].

For more recent distribution learning, structure-explicit approaches, the added complexity of incorporating and representing protein structure brings added challenges. Generative models operating in 3D space should be E(3) equivariant (to translation, rotation, and reflection), typically achieved by augmentation of the training data via random rotation and translation transformations[151,152] or by using equivariant networks on point cloud representations with continuous convolution filters[48] on interatomic distances[153–155]. However, because data augmentation is not an exact solution the model is still susceptible to sensitivity in input orientation. On the other hand, point cloud representations fail to model bond information, typically leading to post-hoc approximation of bond order: an approach that can lead to idiosyncratic bond orders or ligand strain[151,154]. This was significantly improved upon by Peng et al.[156] by also explicitly modelling and predicting bond types in the vector neuron network generative model. In addition to the data being more sparse than the 2D counterpart, often the data is incomplete i.e., none, or just a portion of a co-crystallized ligand is modelled due to poor resolution which may go unnoticed by users, such as PDB codes 1HYZ[157] or 1PL0[158], present in PDBbind[143]. Due to labelled data sparsity, augmenting training data with simulated data[159] (docked ligand poses) is also common practice[151,154,156,160]. This introduces dependence on the method of data augmentation, such as the force-field or other bias induced by a docking algorithm[159]. Moreover, this method of augmentation was originally proposed to augment negative data necessary for discriminative tasks (e.g., pose prediction and binding affinity prediction), therefore, using it as an approach to infer positive data for generative modelling will significantly increase noise due to the inaccuracies of docking and must be used with caution.

There is still no evidence on the best approach given a particular context, as such a meaningful comparison between the approaches classified here is currently lacking. That would require the appropriate standardized evaluation of these approaches where currently no agreed dataset-task-metric exist. Ultimately, prospective validation by synthesis and experimental binding affinity assay is the best method to evaluate structure-based generative model proposals, however, only one Li et al. thus far have conducted prospective validation of a structure-based generative modelling approach[161] (with one of nine structure-based proposals showing weak binding affinity to CDK4, see 1.2.3.7).

1.2.3 Evaluating de novo design methods in the context of drug design

1.2.3.1 Performance metrics

To be able to compare and improve generative model algorithms for *de novo* design, their performance needs to measurable. Many performance metrics have been proposed in the literature that calculate a property based on *de novo* molecules generated from a respective model. Here, these are categorized into measuring intrinsic properties (i.e., without reference any external molecules) and extrinsic properties (i.e., in reference to external molecules, for example, any training data used), and summarized in Table 1.1 and Table 1.2.

Table 1.1: Performance metrics to measure intrinsic properties of *de novo* molecules and hence, generative model performance. Where G^* is a multiset of generated *de novo* molecules including repeated elements, G is a formal set of unique elements, and $(\cdot)_v$ denotes valid outputs only.

Name	Equation	Description
Validity[54]	$\frac{\lvert G_v^* \rvert}{\lvert G^* \rvert}$	The fraction of outputs that correspond to a valid molecular graph.
Uniqueness[54]	$\frac{\lvert G_v \rvert}{\lvert G_v^* \rvert}$	The fraction of distinct valid molecules.
Internal diversity (IntDiv^p)[79,162]	$1 - \sqrt[p]{\frac{\sum_{(i,j)\in G_v \times G_v} sim(m_i, m_j)^p}{\lvert G_v \rvert^2}}$	The average of each molecule's average pairwise Tanimoto distance to all other molecules in the set of valid, unique molecules. Where distance is $1 - sim$, and Tanimoto similarity sim is calculated on the molecules respective ECFP4 fingerprint representation. Two variations exist, IntDiv1 and IntDiv2 depending on p, where IntDiv2 ($p = 2$) takes the root of the average squared similarity.
Sphere exclusion diversity (SEDiv@1k)[147]	$\frac{\lvert g_{v,d} \rvert}{\lvert g_v \rvert}$	The fraction of diverse molecules $(\cdot)_d$ in random sub-sample of 1,000 generated molecules $g \in G$. Where diverse molecules are identified by the sphere exclusion algorithm[163] with a Tanimoto distance cut-off of 0.65. This cut-off corresponds to 80% probability of possessing similar bioactivity.
GuacaMol Filters[54]	$\frac{\lvert \{m \in G_v \lvert filt(m)\} \rvert}{\lvert G_v \rvert}$	The fraction of molecules that pass a filter $filt$ checking for the presence of substructures, where substructure are collated from public and in-house sources[164].
MOSES Filters (MCF & PAINS)[79]	$\frac{\lvert \{m \in G_v \lvert filt(m)\} \rvert}{\lvert G_v \rvert}$	The fraction of molecules that pass drug-like filters $filt$. These are defined as not containing medicinal chemistry (MCF)[79] and PAINS[7] substructures, or physicochemical properties outside the following constraints: allowed atoms (C, N, S, O, F, Cl, Br, H), molecular weight between 250 and 350 Da, logP below or equal to 3.5 and rotatable bonds below or equal to 7.

Table 1.2: Performance metrics to measure extrinsic properties of *de novo* molecules and hence, generative model performance. Where G is a set of valid, unique *de novo* molecules, R is a set of unique reference molecules.

Name	Equation	Description
Novelty[54]	$\frac{\lvert G - R \rvert}{\lvert G \rvert}$	The fraction of unique valid molecules that are not present in a reference dataset. Typically, this refers to the training dataset used to train the generative model.
FCD[165]	$FCD(G,R) = \lvert\lvert \mu_G - \mu_R \rvert\rvert^2 + Tr\left[\sum_G + \sum_R - 2(\sum_G \sum_R)^{1/2}\right]$	The Wassertstein-2 distance between the activation layers of the penultimate layers of ChemNet[166] for the generated set and reference set. Where μ denotes the mean and $\sum$ the covariance of the respective activation layers. This is a proxy measure of the difference in chemical distributions and correlates with differences in drug-likeness, logP, synthesizability, and mode collapse.
Analogue similarity[167]	$\frac{\lvert\{m_g \in G \lvert \exists m_r \in R(sim(m_g, m_r) \geq 0.4)\}\rvert}{\lvert G \rvert}$	The fraction of generated molecules m_g where there exists at-least one reference molecule m_r that is an analogue. Where an analogue is defined as having a Tanimoto similarity sim of 0.4 or more based on ECFP4 fingerprints.
Functional groups or ring systems covered[104]	$\frac{\lvert G_{fg} \cap R_{fg} \rvert}{\lvert R_{fg} \rvert}$ or $\frac{\lvert G_{rs} \cap R_{rs} \rvert}{\lvert R_{rs} \rvert}$	The fraction of either unique functional groups generated fg or ring systems generated rs that are within the reference set, relative to the reference set. Where functional groups are identified as described by Ertl et al.[168], and ring systems are identified by fusion of monocyclic rings if shared atoms exist.
Functional groups and ring systems outside[104]	$\frac{\lvert G_{fg} - R_{fg} \rvert}{\lvert G_{fg} \rvert}$ or $\frac{\lvert G_{rs} - R_{rs} \rvert}{\lvert G_{rs} \rvert}$	The fraction of either unique functional groups generated fg or ring systems generated rs that are within the reference set. Where functional groups are identified as described by Ertl et al.[168], and ring systems are identified by fusion of monocyclic rings if shared atoms exist.
Single nearest neighbour similarity (SNN)[79]	$\frac{1}{\lvert G \rvert} \sum_{i \in G} \max_{j \in R}(sim(m_i, m_j))$	The average maximum similarity of molecules in the generated set to molecules in the reference set. Similarity is calculated as the Tanimoto similarity sim based on ECFP4 fingerprints.
Fragment (Frag) or scaffold (Scaff) similarity[79]	$\frac{\vec{G}_{fr} \cdot \vec{R}_{fr}}{\lvert\lvert \vec{G}_{fr} \rvert\rvert \lvert\lvert \vec{R}_{fr} \rvert\rvert}$ or $\frac{\vec{G}_{sc} \cdot \vec{R}_{sc}}{\lvert\lvert \vec{G}_{sc} \rvert\rvert \lvert\lvert \vec{R}_{sc} \rvert\rvert}$	The cosine similarity between the identically ordered multiplicity of the generated and reference multiset of either fragments fr or scaffolds sc, where $\overrightarrow{(\cdot)}$ denote the multiplicity or count vector. Fragments are identified by BRICS fragmentation[169]. Scaffolds are the Bemis-Murcko scaffolds[170].
Property similarity[79]	$\int_{\mathbb{R}} \lvert F_G(x) - F_R(x) \rvert dx$	The Wasserstein-1 distance between the respective property x distribution of the generated and reference molecules. Where F is the inverse cumulative density function. Previously evaluated properties include logP[171], SAscore[172], QED[173] and molecular weight.

Table 1.3: How simple performance metrics relate to practical relevance, and rule-based or distribution-based generative models. The introduction of distribution-based generative models brings more complex dependencies on data and algorithms.

Property	**Practical relevance**	**Rule-based**	**Distribution-based**
Validity – molecules must adhere to chemical principles e.g., valency.	Critical.	Molecules should always be valid (unless there are systematic errors in building blocks and building rules).	Dependent on molecular representation chosen, complexity of training data and complexity of model.
Uniqueness – the rate at which molecules are duplicated by the model.	Unnecessary if the single *de novo* molecule satisfies all desirable properties.	Dependant on the search algorithm used.	Dependent on the search/optimization algorithm and applicability domain imposed by training data.
Diversity – the scope of chemotypes generated relative to all chemical space.	Unnecessary if *de novo* molecules occupy the most optimal chemical space.	Dependant on the search algorithm and fidelity achievable by chemical building blocks (i.e., atoms, fragments, or reactants).	Dependent on the search/optimization algorithm and applicability domain imposed by training data. May afford greater diversity where rules are difficult to explicitly define (e.g., natural products).
Novelty – the presence of molecules in any training data used.	Critical to fulfil the definition of de novo molecule generation.	Only applicable to seeded models such as genetic algorithms.	Dependent on all model aspects, training data used, molecular representation, architecture etc.
Similarity – the similarity between generated molecules and any training data used.	Unnecessary if *de novo* molecules satisfy all desirable properties.	Only applicable to seeded models such as genetic algorithms.	Dependent on all model aspects, training data used, molecular representation, architecture etc.
Synthetic feasibility – the ability to synthesize a molecule in the lab with relative ease.	Critical for experimental validation and practical application as a therapeutic.	Rules can adhere to known chemical reactions and reaction centres, ensuring a degree of synthetic feasibility (usually to the detriment of diversity).	Synthetic feasibility of molecules may be implicitly learned based on the training data; however, it usually cannot be guaranteed for novel molecules.

In the most basic sense, all generative models must perform well at the task of generating reasonable *de novo* molecules. This means they must firstly generate valid molecules that adhere to chemical principles such as valency. Secondly, generated molecules should be mostly unique (i.e., the model should not re-generate the same molecule over and over). This is irrelevant if the generated molecule is the 'perfect solution', however, this is unlikely and cannot be proved without testing all possible molecules. Thirdly, the generated molecules should be novel with respect to the training data used for distribution-based models – otherwise, a VS could be used instead. However, it is worth noting that the satisfaction of novelty does not guarantee the degree of similarity to any respective training data (where a high degree of similarity may be either desirable or undesirable based on different objectives). Fourthly, it is usually advantageous that generated molecules exhibit a high diversity and do not occupy a very narrow region of chemical space, unless this chemical space is the 'most optimal', which also cannot be easily proved. Lastly, generated molecules should be synthetically feasible i.e., be possible to synthesise in the lab, as molecules will always need to be experimentally tested and if successful, scaled up into production settings. How these principles relate to rule-based or distribution-based generative models is summarised in Table 1.3.

In practice, metrics have also been shown to fail to sufficiently evaluate generative model performance. For example, Renz *et al.*[134] show that a naïve model that samples from a training set and adds a carbon is competitive with 'state-of-the-art' methods according to the simplest metrics of validity, uniqueness and novelty (often the only metrics evaluated). Another difficult property to measure is diversity, which is challenging to combine into a single value in a robust, interpretable way. Internal diversity (described in Table 1.1) takes the average of averages, completely losing the information contained in the underlying distributions. Moreover, it has been known since 1999 that the Tanimoto distance of extended connectivity fingerprints (ECFP) is heavily confounded by the number of heavy atoms[174], and it also possesses rather low resolution at the low-similarity range, due to the low number of shared bits present in this situation. Thus, it is conceivable that internal diversity may also correlate with the number of heavy atoms distribution of a generated set. Sphere exclusion diversity is proposed in an attempt to circumvent these issues and is presented as part of this thesis in Chapter 2. Although many initial performance metrics are flawed when used

individually, the combination of metrics provides a more robust overall picture of generative model performance. Furthermore, more recent improvements have been made in establishing more interpretable metrics, for example, Zhang et al.[104] compared generative models by their ability to produce specific functional groups and ring systems present in GDB-13[32] (see Table 1.2).

1.2.3.2 Benchmarks and benchmark objectives

To be able to compare goal-directed generative models, their performance must be compared on a standardized benchmark objective or task. This enables fair comparison and helps to identify good or poor performance for particular objectives. While the gold standard of measuring model performance is to synthesize and experimentally test *de novo* molecules against the property for which a proxy was optimized (e.g., protein binding assay[17,175] or *in vitro* pharmacokinetic assay), this is intractable considering the experimental resource requirement, number of new generative models that require evaluation, and number of *de novo* molecules proposed by each model (up to billions[104]).

In silico benchmarks exist such as MOSES[79], GuacaMol[54], PMO[55], smina-docking[80], DOCKSTRING[149] and TDC[124]. However, MOSES does not contain any objectives to benchmark goal-directed optimization.

GuacaMol is seminal benchmark that contains a suite of 20 objectives for goal-directed optimization. Each of these objectives evaluates the similarity to a reference molecule or set of reference molecules. Four generative models were compared including Graph GA[56], Graph MCTS[56], SMILES GA[176] and SMILES LSTM (with HC)[122]. Overall, Graph GA and SMILES LSTM achieved excellent performance, with only four tasks achieving a score below 0.8 (out of 1). Additionally considering the type of chemistry generated with GuacaMol filters (see Table 1.1), the SMILES LSTM was vastly superior, as comparatively half the Graph GA molecules passed filters. However, this excellent performance also highlighted that the objectives were not difficult enough, especially considering leeway for more performant generative models. This was also mentioned by the authors. Furthermore, the tasks proposed are not very relevant to real drug design challenges, for example, binding affinity. A recent extension to this benchmark, PMO, additionally accounts for sample efficiency of objective optimization. A significant contribution of this benchmark was the implementation of

25 generative models, a much broader comparison than four. This work also reinforced the performance of SMILES-based language models as the best performing models.

More recent, docking-based, benchmarks have also emerged including smina-docking, DOCKSTRING and TDC. Using docking as a proxy for binding affinity is a more relevant objective to drug design. However, these docking benchmarks only consider 1-4 protein targets and do not appropriately prepare *de novo* molecules *via* stereoisomer enumeration, tautomer enumeration and protonation. Moreover, docking algorithms tend to over score large, greasy molecules which isn't traditionally a problem as these are filtered out of VS compound libraries. Therefore, this objective is heavily confounded by a generative model's ability to generate these compounds which is usually an undesirable characteristic – these objectives must be more carefully defined to avoid this behaviour.

Beyond established benchmarks, other objectives have been proposed and used to compare generative model performance *via* the highest score achieved by de novo molecules. For example, penalized logP[106] (logP penalized by SAscore and number of rings). Not only is the task practically irrelevant to the drug design process, but the evaluation only accounts for a model's ability to optimize an arbitrary function. Therefore, this evaluation is only meaningful if the scoring function has been proven robust, accurate and validated in that area of chemical space (which is often not the case).

1.2.3.3 De novo design objectives and their relevance to drug design

Even if a generative model generates a diverse array of valid, novel, and synthetically feasible molecules; or the ability to recover or generate similar molecules to known drugs, does not guarantee that generated molecules are relevant to the objective of identifying drug candidates. For this, molecules must possess other necessary on-target, pharmacokinetic, practical, and legal properties, as discussed in 1.1. Some of the properties required, how they might be approximated by scoring functions to guide goal-directed generative models, and their associated limitations is discussed in Table 1.4.

Table 1.4: Proxy functions, and pitfalls of such approaches, that can be used for goal-directed *de novo* design based on the properties required for molecules to be considered for lead optimization or drug candidate selection, as outlined in 1.1.

Property	**Proxy scoring function**	**Pitfalls**
On-target activity – a molecule must exert the desired effect on a respective biological target.	• 2D/3D similarity to known active molecules • QSAR models • Proteochemometric models • Docking simulations • Free energy perturbation (FEP)	• Active ligands are not always known, restricting the application of similarity and QSAR/ML approaches to novel targets. • QSAR models and docking simulations predict on-target affinity which does not always translate to bioactivity (e.g., polypharmacology required, incorrect binding mode, sub-optimal binding kinetics, metabolite activity etc.)[20]. • Similarity approaches (and the limited applicability domain of QSAR models) restrict novelty and diversity of de novo molecules[75]. • Different QSAR models with similar performance metrics behave differently in prospective use cases[177]. • Docking simulations can be inaccurate[141,142] and highly target dependent[178]. • Molecular dynamics based simulations like FEP are too computationally expensive.
Novelty (legal) – a molecule must not infringe on competing intellectual property.	• Dissimilarity to patented molecules • Substructure filters	• Patented molecules are typically enumerated Markush structures and are thus more difficult to explicitly define for similarity-based approaches – may result in missed similarities e.g. Walters et al.[18]. • Computationally expensive to exhaustively check similarity to all patented molecules.
Synthetically feasible – a molecule must be possible to efficiently synthesized and ideally at scale.	• Heuristics (e.g., SAscore[172]) • Machine learning models (e.g., RAScore[179]) • Computer-aided synthesis planning (CASP)	• Single numerical values returned only increase the probability of possible available synthetic routes – i.e., synthesis is not guaranteed by finding a validated synthetic route they usually provide no interpretation of synthetic routes e.g., number of steps (although modelled in SCScore[180]), cost, availability of starting materials etc. • CASP models are too computationally expensive.
Pharmacokinetics – a molecule must be able to reach the target/s	• Heuristics (e.g., number of rotatable bonds, Crippen LogP[171])	• Heuristics are fast becoming outdated, with exceptions to popular rules such as beyond rule of five drugs[181,182].

of interest (ideally *via* traditional administration routes e.g., oral).	• QSAR/QSPR models	• QSAR/QSPR models are typically trained on data labelled using in vitro assays or simplified cellular systems, failing to account for the complexity and heterogeneity of organs and tissues that are adaptive, unique to individuals, and dependant on the microbiome[183].
Suitable off-target profile – a molecule or its metabolites must not exert an effect on off-targets.	• (Dis)similarity to active off-target molecules • QSAR models	• Off-targets or respective ligand data is not always known *a priori*. • QSAR model accuracy contains a heavy bias towards targets that contain more data[184]. • Similarity to known off-target ligands is a very difficult optimization task (due to the similar nature of protein families and subtypes, ergo similar ligands) - e.g., 40% of 112,000 kinase inhibitors exhibit multi-kinase activity[185]. • Aggregate similarity across many off-targets ligands will result in a difficult Pareto optimization problem (i.e., will not always achieve dissimilarity to all off-targets).
Toxicity – a molecule or its metabolites must not induce any toxic side-effects.	• Substructure filters • QSAR models	• Toxicity data is expensive to acquire and thus typically comprises smaller datasets - e.g., one of the largest (Tox21) which contains annotations for ca. 10,000 compounds[186]. This makes it more difficult to apply modern neural network models that rely on large quantities of data[187]. • Toxicity is also heavily dependent on dose[188], increasing the prediction challenge for QSAR models. • Substructure filters can be effective but crude, for example, discarding substructures in which toxicity could be mitigated with design changes such as nearby sterically hindering groups to avoid site metabolism (e.g., reducing aldehyde oxidase metabolism[189,190]).

The scoring functions used to evaluate *de novo* molecules and hence guide goal-directed generative models come with their own limitations, as discussed in Table 1.4. A common theme in scoring function pitfalls, that is also relevant outside their use in generative models, is the large disconnect between respective property endpoint and scoring function proxy (where a proxy is a function that should represent the endpoint property). For example, the biological gap from biological efficacy (e.g., reduced disease progression) to on-target activity (e.g., binding affinity and mode of action) to implemented proxy (e.g., a QSAR model that predicts the pIC_{50} of a single protein binding assay). Firstly, this measure is only a proxy for on-target activity due to the many complex dependencies not captured, such as, effect size (e.g., percent inhibition), mode of action (specific residue interactions that may be required to exert specific downstream effects), binding kinetics (e.g., desirable range can be influenced by protein turnover), as well as co-dependency on other properties such as bioavailability. Let alone, the biological gap from on-target activity to efficacy which depends on many more factors downstream of on-target activity and can even be highly dependent on individuals. The closest known attempt to moving closer towards phenotype and hence bridging this biological gap was implemented by Méndez-Lucio et al., where the generative model was trained to generate molecules conditioned on gene expression profiles (downstream of on-target activity) using generative adversarial networks[111], as well as a similar approach using cell morphology[191]. Although these approaches somewhat close the biological gap, the information incorporated tends to be much noisier (and hence the signal is much more difficult to identify), introducing new confounding variables such as different cell lines. Moreover, there is often no clear mechanistic link to actual endpoints such as efficacy or toxicity. Ultimately, limited quantity and availability of data here is a significant limiting factor.

Further limitations not discussed in Table 1.4 include the dependant behaviour of generative models when optimizing the values returned by specific scoring functions, which can exacerbate scoring function pitfalls. Recently Renz et al. observed the sensitivity of generative models towards generating molecules optimal specifically against a QSAR model data split or hyperparameter setting[134], demonstrating the effect of limited QSAR model robustness and applicability domain on *de novo* molecule generation. This was further characterized by Langevin et al.[135]. As suggested by Olivecrona et al. “any predictive model to be used in conjunction with

the generative model should cover a broad chemical space within its domain of applicability, since it initially has to assess representative structures of the dataset used to build the Prior"[76]. In other words, at some point during training, the model is likely to evaluate molecules outside its domain of applicability resulting in aberrant predictions. This increases the importance of predictive model robustness and confidence prediction. Overall, the dependant relationship between generative model and scoring function seems to adhere to Goodharts' law[192]: "When a measure becomes a target, it ceases to be a good measure".

1.2.3.4 Evaluation in the context of data availability and project priorities

It is important to consider the context of prospective application when evaluating generative models, as they must be applicable to real-world drug design problems.

Projects aimed to drug novel targets (a.k.a. first-in-class) will have no reported ligands to utilise for transfer learning or to train an ML-based scoring function. One way of approaching this is to use structure-based scoring functions if a protein structure is available. More and more generative models utilise 3D structure or docking-based scoring functions that can be applied in this scenario (see 1.2.2.6); however, so far only one has been experimentally validated.

Another common prospective application may include design of better drugs than are currently available on the market (a.k.a. best-in-class), whereby known ligand data is more likely to be available for transfer learning or scoring function training. However, proposed molecules must be novel enough to not infringe on competing intellectual property – which may be difficult to attain if relying on known, potentially patented ligand data. Moreover, success will likely depend on an overall better profile considering all of the properties discussed in Table 1.4, where an optimal multi-parameter profile is most advantageous. However, multi-parameter optimization (MPO/MOO) results in a Pareto optimization problem, where optimization of one property is to the detriment of another. The ability to optimize many desired properties simultaneously is one of difficulty and often neglected in generative model publications receiving relatively little attention[61,102].

Overall, model evaluation must focus more effort on the context of prospective application, so that evaluation is more interpretable when considering integration of *de*

novo design with generative models into real-world projects. For example, novelty with respect to existing reported ligands or existing literature is almost never considered.

1.2.3.5 Evaluation in the context of traditional in silico methods

Understanding how generative models compare to other traditional and non-AI methods like VS has not yet been seriously considered. Recently Steinmann et al.[57] used a genetic algorithm to find *de novo* molecules with good docking scores, comparatively their approach identified 1.9x more high scoring molecules than conventional VS; however, there approach required to dock 1.6x more molecules than the VS. Therefore, the convenience-enrichment trade-off likely falls in the favour of VS based on these results, as the screening library is commercially available without the need for synthesis. Furthermore, comparison to other techniques such as bio-isostere replacement based on medicinal chemistry precedent[18,193] would further the interpretation of generative model performance.

1.2.3.6 Evaluation of a hypothetical de novo design scenario to demonstrate evaluation challenges

To illustrate the challenges of evaluating generative model performance in the context of drug design with the performance metrics and objectives available, a hypothetical *de novo* design scenario is discussed. Figure 1.14 illustrates a hypothetical chemical space occupied by either literature reported bioactive molecules, *de novo* molecules proposed by a generative model optimized to maximize a QSAR model prediction (trained on respective bioactive molecules), and a training dataset. The hypothetical aim is to identify new bioactive chemical series, akin to real world drug discovery. All *de novo* molecules are assumed valid and synthesizable.

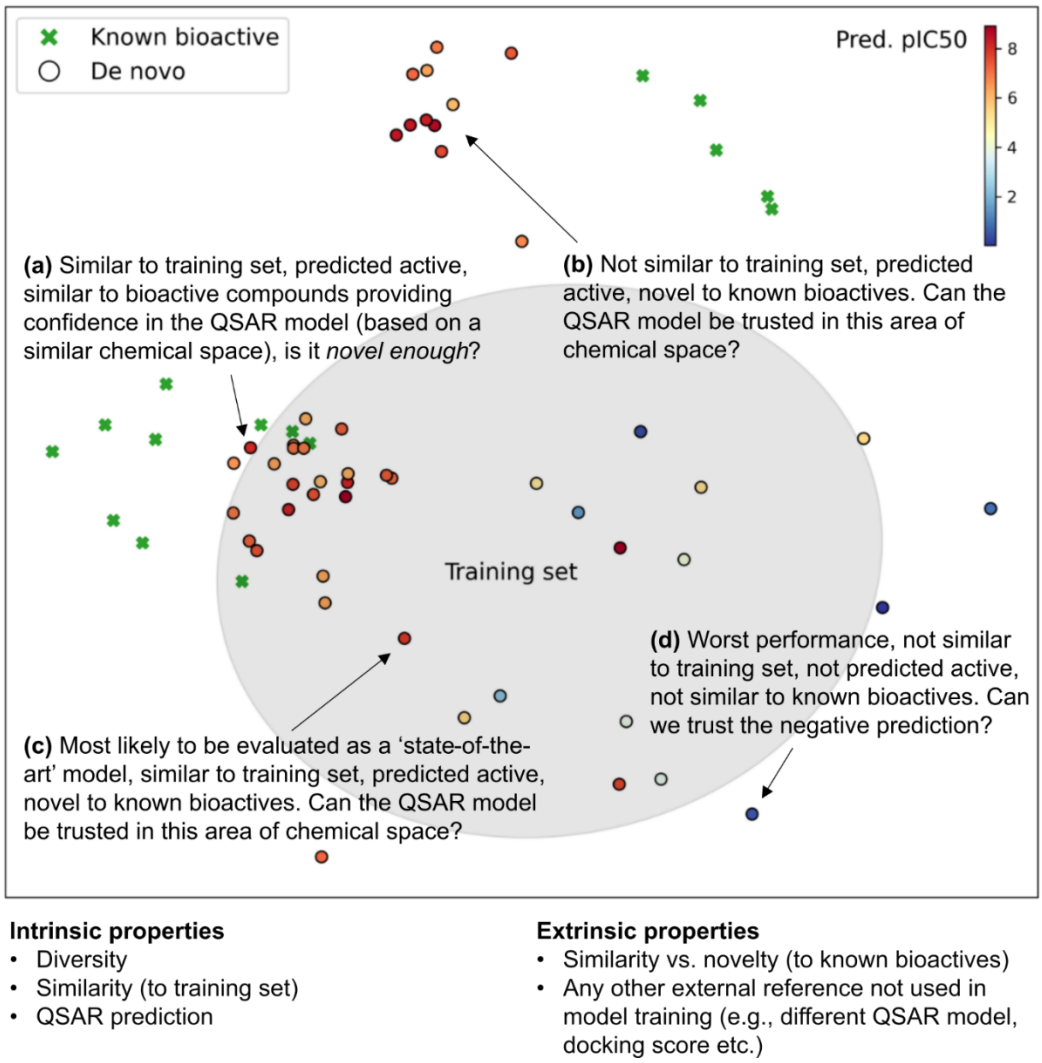


Figure 1.14: Hypothetical evaluation of *de novo* molecule chemical space, optimized to maximize predicted binding affinity. Schematic of chemical space occupation by known bioactive molecules, *de novo* molecules, and training data. The training dataset for the generative model is shaded grey. Certain regions that *de novo* molecules occupy have been annotated to reflect considerations when evaluating model performance.

Figure 1.14 highlights the difficulty comparing, for example, *de novo* molecules at Figure 1.14a and at Figure 1.14b**.** Where molecules at Figure 1.14a perform well considering close similarity to the training set, high predicted pIC50 and similarity to known bioactives (providing confidence in the QSAR predictions). However, the molecules may be too similar to known bioactives and therefore, not novel enough to be of use in practice (i.e., avoid competing intellectual property). Furthermore, these molecules could likely be found using traditional drug design approaches (e.g., decorating scaffolds and/or scaffold hopping), raising further concerns as to the real world benefits of generative models over traditional methods. This situation is similar to that reported by Zhavoronkov et al.[17], where a *de novo* molecule was later found to

be very similar to a marketed drug (and similar to a molecule in the training set)[18], it has further been speculated that if a similar discovery was made via more traditional techniques it would not even be novel enough to report[18]. If the molecules at Figure 1.14b are now considered**,** the generative model would be considered poorly performing due to dissimilarity to the training set, even though molecules are predicted active and are novel enough with respect to known bioactives. It raises the question: if the molecules do satisfy optimized property requirements (i.e., predicted bioactive), does it matter that they are dissimilar to the training set? Secondly, now the molecules are in a new region of chemical space, how much can we trust the QSAR predictions? These are usually unanswered questions that are highly context dependent. By using the most common performance metrics, it would be a model that results in molecules at Figure 1.14c that would be considered 'state-of-the-art' due to a highly optimized predicted activity and similarity to the training set – with complete disregard to the applicability of such a QSAR model in that region of chemical space, or novelty with respect to known bioactives. This hypothetical scenario is designed to stress the importance of measuring performance with respect to external references which are more meaningful in practice, despite being difficult to measure. This also stresses the importance of understanding a scoring functions applicability domain in the context of evaluation.

1.2.3.7 Prospective evaluation

Compared to the number of recent ML-based generative model algorithms proposed, relatively few have undergone any kind of prospective evaluation by experimental testing of *de novo* molecules. Merk et al.[194] used transfer learning with an LSTM model and tested four *de novo* molecules on RXR potency, two of which had double-digit micromolar activity. Zhavoronkov et al.[17] used REINFORCE with a VAE model and tested six *de novo* molecules on DDR1 inhibition, three of which had sub-micromolar activity and one with micromolar activity. Li et al.[161] used transfer learning with an LSTM model and tested 12 *de novo* molecules on PIM1 and CDK4 inhibition, one of which had sub-micromolar PIM1 activity and three of which had micromolar CDK4 activity. Yang et al.[195] used transfer learning with an LSTM model and tested one *de novo* molecule on p300/CBP HAT inhibition, which displayed nanomolar activity. Grisoni et al.[175] used transfer learning with an LSTM model and tested 28 *de novo* molecules on $LXR_{\alpha/\beta}$ activation, twelve of which had micromolar activity. Hua et al.[196]

used a GRU model to help build a combinatorial library which was subsequently virtually screened, they then tested 17 *de novo* molecules on MERTK inhibition, 10 of which had sub-micromolar activity and 5 of which had micromolar activity. Lastly, Moret et al.[197] used transfer learning with an LSTM model to build a focussed VS and tested 16 commercially available *de novo* molecules on PI3Kγ binding (one of which exhibited sub-micromolar activity), two non-commercially available *de novo* molecules (both of which exhibited nanomolar activity), and four human-derived analogues (all of which exhibited sub-micromolar activity).

However, only Grisoni et al.[175] was evaluated objectively, with the exception of just 6/51 not selected for synthesis due to price. All other prospective evaluations discussed above either had human influenced selection of *de novo* molecules for testing[17,161,194–197] or underwent human-designed modifications of *de novo* molecules[161,197]. For example, Moret et al. tested the 12/1,121,735 *de novo* molecules that were commercially available, which is clearly an extremely small sample size relative to possible molecules. Even the 2/47 *de novo* molecules that were selected for testing by high bioactivity prediction were human selected, again providing a small sample size to represent objective model performance. Alternatively, Zhavoronkov et al.[17] generated 30,000 *de novo* molecules that were expertly filtered and selected to just 40 compounds (6 of which were synthesized). This leads to difficult delineation between performance of the generative model or the proceeding VS and expert filtering. Although it is relevant how best to select from (potentially millions of) *de novo* molecules, this does confound objective evaluation of generative model performance. Clearly defining levels of automated chemical design[198] should help to clarify human influence moving forward.

1.3 Aims & Objectives

The field of *de novo* design is rapidly undergoing advancement due to the introduction of distribution-based, generative ML models in recent years. Over the course of these doctoral studies more than 100 *de novo* drug design methods have been published, and the AI field has solved grand challenges such as predicting 3D protein structures to unprecedented accuracies[69]. However, many challenges still exist for generative models relating to their performance evaluation and application in drug design, especially in the larger context of real drug discovery scenarios. The aims of this thesis

were to investigate and improve the application of goal-directed generative models to drug design by additionally incorporating structure-based principles.

1. The inconsistent comparisons in generative model performance by either optimizing different or arbitrary, irrelevant objectives makes it difficult to discern well performing generative models. In an attempt to help address this, Chapter 2 introduces MolScore; an open-source python software to facilitate the easy application of configurable, drug design relevant objectives for use with any generative model. In addition, this software compiles performance metrics proposed in the field, as well as two graphical user interfaces to improve user friendliness. It contains enough functionality to be utilized for real-world drug design projects in combination with generative models. Ideally, this can be adopted by researchers in *de novo* design to provide a standardized framework for objective design and thus, provide more standardized comparison between generative model performance.

2. To leverage the advantages of SBDD principles over LBDD and investigate the effect of alternative scoring functions, MolScore was used to configure a docking-based or ML-based ligand classification scoring functions. These were combined with a published chemical language model, REINVENT[76], that uses RL to optimize molecule generation to maximize an objective. Chapter 3 describes the results of this first comparison between SBDD and LBDD scoring functions for generative models, and how using SBDD *via* docking can improve the overall performance of the generative model, as well as increase the coverage of known bioactive chemical space.

3. A major caveat of using molecular docking simulations as goal-directed scoring functions is large computational expense. Therefore, the next logical step is to choose to improve either the efficiency of *de novo* molecule optimization, or the efficiency of structure-based molecular evaluation. Chapter 4 outlines how that choice was addressed by introducing Augmented Hill-Climb as a hybrid RL algorithm to improve the efficiency of *de novo* molecule generation. This increased the speed of optimization ~45-fold and the optimization ability ~1.5-fold compared to the baseline REINVENT when optimizing the docking score of *de novo* molecules, but crucially maintained desirable distributions over the chemical space sampled.

Chapter 2: MolScore: A scoring and evaluation framework for *de novo* drug design

2.1 Introduction

Performance measurement for the comparison of generative models is important to be able to quantify the most impactful improvements for future research. Many performance measures and benchmarks have been proposed to measure generative model performance. However, these often fall short in the relation to their prospective use case requirements. Moreover, there is often a lack of consideration for the quality of chemistry generated[82], many models are still applied to prospectively irrelevant objectives (such as rediscovery[54] or penalized logP[106]), and scientific significance of proposed *de novo* designs is often overlooked[18]. This is understandable due to the sheer number of approaches not all models can be prospectively validated, that many new methods stem from the computer science and machine learning domain where drug design expertise may be lacking and therefore simple, easy-to-implement objectives are preferred, and finally that benchmarks are still needed to compare approaches.

In addition to the benchmarks discussed in 1.2.3.2, there exists more general software for objective design for goal-directed generative models, without necessarily quantifying a performance outcome. REINVENT[76,99,148], one of the seminal approaches, implements a suite of configurable scoring functions for use with its respective goal-directed generative model architecture. However, the design of the source code is deeply integrated with the REINVENT generative model and therefore it is not straightforward to be used interchangeably with other goal-directed generative models for standardized comparison. Another framework, the Therapeutic Data Commons (TDC) platform[124], reimplements the GuacaMol suite (with some customizability, e.g., reference molecule) and provides several additional capabilities such as docking, three synthetic accessibility scores, three molecular descriptors and three pre-trained activity models. However, not all scoring functions are customizable and score transformation or aggregation (for use in a multi-parameter setting) must be manually coded – introducing a problem with respect to standardization and reproducibility across users.

MolScore introduced in this chapter aims to address some of these frustrations by providing an open-source suite of configurable objectives and evaluation metrics designed for use with any generative model. MolScore can be used to design multi-parameter objectives for practical use in real-world drug design coupled with a generative model of choice. While MolScore is not in itself a benchmark suite, it can be used to share and run standardized objectives that may be proposed as challenging benchmark tasks for the field. In addition, MolScore contains two graphical user interfaces (GUIs) to aid user experience in both writing configuration files and analysing generated *de novo* molecules.

A high-level comparison of MolScore compared to existing software/benchmarks solutions is shown in Table 2.1. MolScore can easily reimplement any of the GuacaMol[54] benchmark objectives and contains all performance metrics proposed by MOSES. MolScore contains similar functionality to conduct docking *via* interaction with a variety of docking software but crucially also contains some appropriate ligand preparation protocols that handle stereoisomer numeration, tautomer enumeration and protonation states. In contrast to REINVENT, MolScore is designed to plug-and-play with different generative models. Lastly, MolScore contains a greater number of scoring functions compared to TDC, with each one more configurable. Moreover, MolScore handles multi-parameter configuration *via* the configuration file thereby standardizing transformation and aggregation.

Table 2.1: Comparison of MolScore to software and benchmarks for *de novo* molecule generation.

	Fixed / configurable [a]	**Goal-directed objectives**	**Evaluation metrics**	**Generative model agnostic [b]**	**Graphical user interface**
GuacaMol	Fixed	✓	✓	✓	
MOSES	Fixed		✓	✓	
PMO	Fixed	✓		✓	
Smina-docking	Fixed	✓		✓	
PyTDC	Fixed	✓		✓	
DOCKSTRING	Fixed	✓		✓	
REINVENT (+DockStream)	Configurable	✓			✓
MolScore	Configurable	✓	✓	✓	✓

[a] Configurable without having to write any code to design the objective.

[b] Easily implementable for most generative models.

2.2 Materials and methods

MolScore is an open-source software written in Python 3, published under an MIT license and distributed *via* GitHub (https://github.com/MorganCThomas/MolScore). It depends on several packages such as RDKit[199], PyTorch[200], Streamlit, as well as integrating published works in the field such as RAscore[179], AiZynthFinder[201] and ChemProp[202]. MolScore is split into two sub-packages: 1) `molscore` for scoring *de novo* molecules proposed by a generative model, and 2) `moleval` for *post-hoc* evaluation using a suite of evaluation metrics. The structure of the sub-packages can be seen in Figure 2.1 and integration into goal-directed *de novo* molecule generation in Figure 2.2. The following sections provide details of each sub-package.

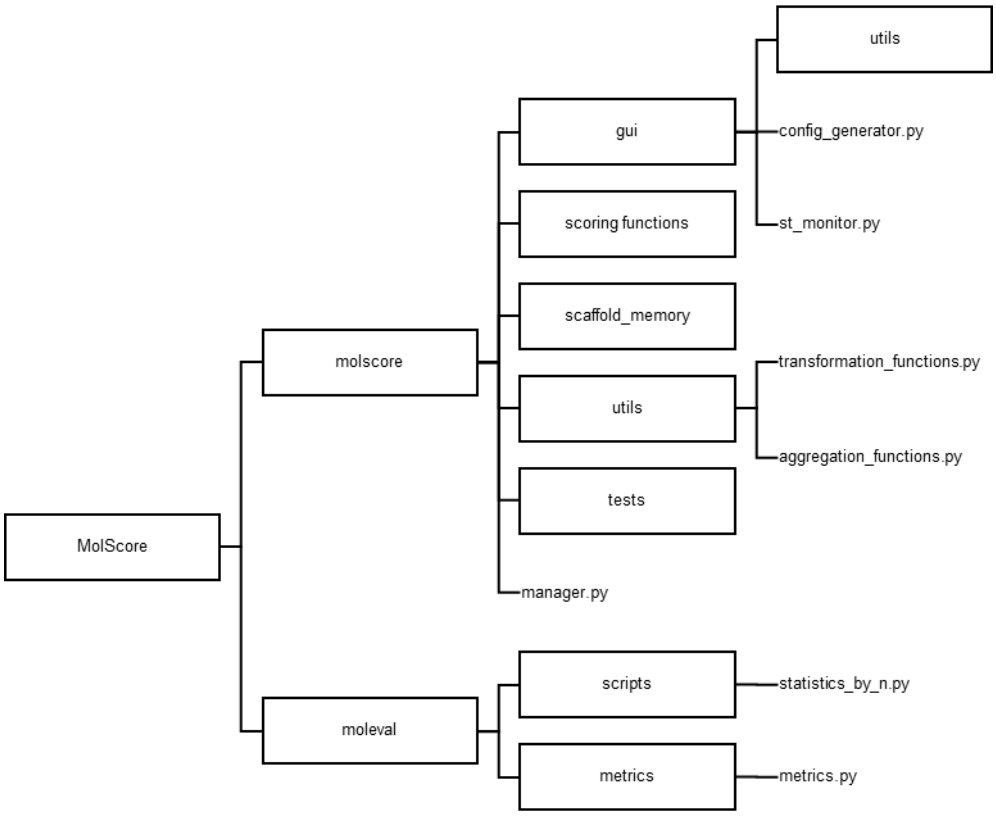


Figure 2.1: Design of the `molscore` and `moleval` sub-packages. The main elements of `molscore` include the `manager.py` module that interacts with a generative model and manages all aspects scoring molecules according to the objective, the `gui` folder that contains the scripts to aid in the setting up of configuration files and monitoring/analysis of *de novo* molecule, the `scoring_functions` folder contains modules for individual scoring functions used, `scaffold_memory` contains code that defines the diversity filters, and `utils` modules that contain the code for the transformation and aggregation functions. The main elements of the `moleval` package are the metrics.py module that computes evaluation metrics and the `statistics_by_n.py` script that computes the evaluation metrics to a `molscore` output file every *n*-steps or *n*-samples.

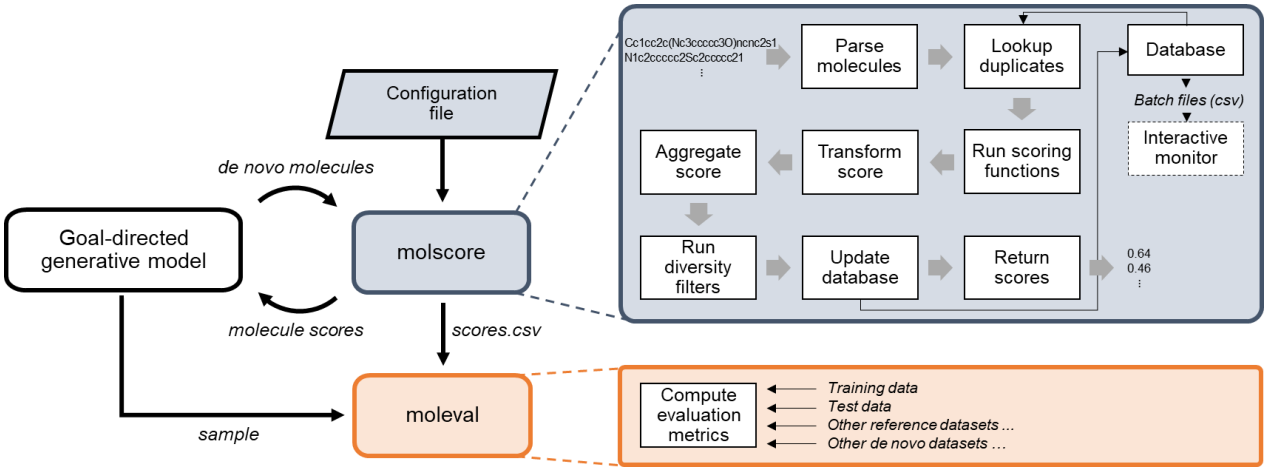


Figure 2.2: Schematic representation of `molscore` (and `moleval`) Python packages and integration with a goal-directed generative model. `molscore` takes as input a configuration file describing the user-defined objective, and iteratively scores *de novo* molecules during the course of an optimization run. `moleval` can then be used to compute a suite of performance metrics to evaluate the *de novo* molecules generation.

2.2.1 molscore

The first sub-package, `molscore`, handles how to score *de novo* molecules. It is a collection of scoring functions, diversity filters, transformation functions and aggregation functions that can be used interchangeably, all managed by a central module `manager.py` (see Figure 2.1). This module contains a python class called `MolScore` that is initialized with a configuration file defining how to score molecules. Once initialized, it takes as input a list of molecules (in SMILES representation) and returns a list of their respective scores as output, designed to be repeated in an iterative fashion (e.g., steps/epochs) over the course of a run. During each iteration, several intermediate steps are taken, as visualized in Figure 2.2. First, molecules are parsed to check for validity (by parsing with RDKit), their SMILES are canonicalized and intra-batch uniqueness is checked. Inter-batch molecule uniqueness is then cross-referenced with previously generated molecules within the run and if the molecule was previously generated its previous score can be reused. This can save valuable time if long-running scoring functions are being used (for example, molecular docking) and if a generative model is susceptible to generating the same molecules multiple times. User-defined scoring function(s) are run only for valid and unique molecules with invalid molecules being assigned scores of 0 and duplicated molecules being assigned their score from a single unique representative. Scores returned by any scoring functions run are then transformed and aggregated according to the user definition and diversity filters applied to penalize the final score of non-diverse molecules, if

chosen. The final results are then added to the run record. In addition, a CSV file is output for each iteration in the run, allowing a GUI to analyse intermediate results during the course of a run. Finally, when the run has concluded, a CSV file is written to the output directory with a full record of molecules generated and their respective scores.

A broad array of functionality is available to define an objective, as outlined in Table 2.2. In summary, the suite of scoring functions includes physicochemical descriptors, 2D and 3D molecular similarity to reference molecules, substructure matching, use of Scikit-Learn[203] prediction models including trained bioactivity models on approximately 2,300 on ChEMBL31[204] targets with PIDGINv5[205], interfacing with five docking software coupled with four available ligand preparation protocols, and finally three synthetic accessibility measures. Most scoring functions are parallelisable using Python's built-in `multiprocessing` module, while longer running scoring functions such as docking and ligand preparation can be parallelised using `Dask`[206] to allow parallelisation over a compute cluster.

Table 2.2: Functionality available within the `molscore` sub-package.

			License required	References
Scoring functions	Descriptors	RDKit Descriptors	No	199
		Penalized logP	No	106
		Maximum consecutive rotatable bonds	No	
	Similarity	Isomer similarity	No	54
		Fingerprint similarity	No	54
		Molecular substructure match	No	54,76
		Molecular substructure filters	No	76
		ROCS	Yes	207
		Open 3D Align	No	208
	Applicability domain	Maximum similarity	No	209
		Feature range	No	209
		Physchem range	No	209
	Predictive models	Scikit-learn models	No	203
		PIDGINv5	No	205,210
		ChemProp	No	211
	Docking	Glide	Yes	212
		PLANTS	Yes	213
		GOLD	Yes	214
		OEDock	Yes	215

		Smina	No	[216]
	Synthesizability	SA score	No	[172]
		RA Score	No	[179]
		AiZynthFinder	No	[201]
Scoring function utilities	Fingerprints	ECFP (Morgan), Atom-pair, Topological-torsions, MACCSkeys, RDKit, Avalon, Pharm2D	No	[199]
	Similarity measure	Tanimoto, All bit, Asymmetric, Braun Blanquet, Cosine, McConnaughey, Dice, Kulczynski, Russel, On bit, Rogot Goldberg, Sokal	No	[199]
	Molecule preparation pipelines	GypsumDL	No	[217]
		Ligprep	Yes	[218]
		Epik	Yes	[219]
		Moka	Yes	[220]
Diversity filters		Unique	No	
		Occurrence	No	
		IdenticalMurckoScaffold	No	[167]
		IdenticalTopologicalScaffold	No	[167]
		CompoundSimilarity	No	[167]
		ScaffoldSimilarityAtomPair	No	[167]
		ScaffoldSimilarityECFP	No	
Transformation functions		Normalize	No	

		Linear threshold	No	[54]
		Gaussian threshold	No	[54]
		Step threshold	No	
Aggregation functions		Weighted sum	No	
		Auto-weighted sum	No	[221]
		Product	No	
		Weighted Product	No	
		Auto-weighted product	No	[221]
		Geometric Mean	No	
		Arithmetic Mean	No	
		Pareto front	No	[221]

2

2.2.1.1 Scoring functions

Molecular descriptors

This scoring function calculates a selected range of molecular descriptors available in RDKit[222] including QED[173], SAscore[172], CLogP[171], molecular weight, heavy atom count, heavy atom molecular weight, number of H-bond acceptors, Number of H-bond donors, number of heteroatoms, number of rotatable bonds, number of aromatic rings, number of aliphatic rings, number of rings, topological polar surface area (TPSA), formal charge, molecular formula and Bertz complexity [223]. Another non-standard descriptor calculated that is commonly used as a proxy optimization objective is penalized logP[106], where logP is penalized by the synthesizability as measured by the SAscore ($SA(m)$) and the number of rings with six or more atoms ($cycle(m)$). In addition, the maximum number of consecutive rotatable bonds is calculated as a proxy to identify highly flexible molecules. This is calculated by first identifying rotatable bonds i.e., any two non-terminal atoms joined by a single bond where at least one atom isn't within a ring using the SMARTS pattern [*!R!D1]-[*!D1] (amides and esters are excluded by removing atoms matching [NX3][CX3](=[OX1]) and [OX2][CX3](=[OX1]) respectively). Single rotatable bonds between different rings are also identified by default, however, these will only ever constitute a maximum bond length of one. Once all rotatable bonds are identified, consecutive rotatable bonds are identified by linking atoms via RDKit. Note that branching is ignored.

$$Penalized\ logP(m) = logP(m) - SA(m) - cycle(m)$$

Molecular similarity

Several scoring functions are available to score molecules based on molecular similarity to a reference molecule or a set of reference molecules.

Isomer similarity as implemented in GuacaMol by Brown et al.[54] calculates the isomer similarity to a user-specified reference molecule based on its molecular formula. More concretely, molecules are scored based on the geometric mean of the Gaussian transformed distance to each element, and the total number of elements. I refer the reader to Brown et al. for further details.

Fingerprint similarity scores molecules based on their molecular similarity to a user-specified reference molecule or set of reference molecules based on their

respective fingerprints. Any fingerprint and similarity measure shown in Table 2.2 can be used. If multiple reference molecules are specified, molecules can either be scored based on their maximum or mean similarity. If a user-specified similarity threshold is provided, then molecules are scored based on the fraction of reference molecules with a similarity above that threshold.

Molecular substructure match scores molecules based on whether they contain a user-specified substructure or set of substructures defined using SMARTS patterns. If multiple substructures are specified, molecules can either be assigned a score of 1 based on matching either any substructure or all substructures.

Molecular substructure filters is the reverse of substructure match, assigning a score of 0 to molecules if they contain any user-specified substructures. Pre-set lists of SMARTS are provided including AZ[76], PAINS[7] and MCF[79].

ROCS scores molecules based on their 3D similarity to a user-specified reference 3D molecule as calculated using OpenEye's ROCS software[224]. As scored molecules don't contain 3D information, conformations are generated by OpenEye's Omega[225].

Open 3D Align scores molecules on their 3D similarity to a user-specified reference molecule or set of reference molecules as calculated using Open3DAlign[208] as implemented in RDKit. Reference molecules are pre-processed such that if they do not contain a 3D conformation, conformations are generated using RDKit, and conformations are aligned to reference molecules that do have specified conformations – the conformation with the closest alignment to a reference molecule is selected. If no reference molecules contain conformations, conformations are generated for all molecules using RDKit and the first molecule undergoes a full pairwise alignment to all other reference molecule conformations – the conformation with the best average alignment to other reference molecules is selected and the remaining are aligned to that reference as before. Scored molecules also undergo conformation generated by RDKit and are then aligned to each reference molecule using Open3DAlign. The score can either be based on the maximum, minimum, mean or median similarity to reference molecules. Additionally, a pharmacophore fingerprint is generated based on the 3D conformation similar to Jung et al.[226].

Applicability domain

Langevin et al.[209] recently proposed applicability domain filters to help control the chemical space of generated molecules with respect to a reference set of molecules. Here I reimplement these as scoring functions to score molecules based on their applicability domain.

Maximum similarity scores molecules based on their similarity to reference molecules. This can be directly re-implemented using the 'fingerprint similarity' scoring function previously described.

Feature range assigns a score of one to molecules if all their features are within the range of the reference molecules and zero otherwise. Features are calculated as fingerprint bits, where any fingerprint in Table 1 can be used. For example, if a molecule contains a fingerprint bit not identified anywhere in the reference molecules it will be assigned a score of 0, alternatively, if it does not contain a fingerprint bit that is present in every reference molecule it will be assigned a score of 0.

Physchem range assigns a score of one to molecules if all their physicochemical properties are within the range of reference molecules and zero otherwise. Physicochemical properties calculated include QED, number of H-bond donors, number of H-bond acceptors, number of rings, number of rotatable bonds, TPSA, logP, molar refractivity, molecular weight, fraction of SP^3 carbons, heavy atom count, fraction of Bemis-Murcko scaffold heavy atoms, size of larger ring, size of smallest ring, total charge, number of positive charges, number of negative charges, and number of chiral centres.

Predictive models

Scikit-learn models scores molecules based on loading a user-specified Scikit-Learn[203] predictive model. Either a classifier or regressor model can be used based on the assumption that the 'predict_proba' or 'predict' method is used respectively. Moreover, fingerprints specified in Table 1 can be used for featurization, if the training molecules were featurized differently, then a custom scoring function will need to be implemented to ensure molecules are featurized accordingly.

PIDGINv5 scores molecules using any of 7,468 pre-trained random forest classifiers based on 2,734 ChEMBL protein targets if they have sufficient training data (more than 10 active molecules) at different activity thresholds (1,341 at 0.1 μM, 1,704

at 1 µM, 2,086 at 10 µM and 2,337 at 100 µM)[205]. All models were trained following a previously published approach[210] on the latest version of data available in ChEMBL31 and PubChem (data accessed December 2022). Prediction IncluDinG INactivity (PIDGIN) uses sphere excluded data from PubChem to augment inactive datasets for ChEMBL targets where few known inactives exist. Test results based on 5-fold stratified scaffold split are shown in Figure 2.3. I refer the reader to Mervin et al.[210] for further methodological details that were followed in this latest version.

ChemProp scores molecules based on loading a user-specified ChemProp[211] predictive model – a popular implementation of message-parsing neural networks for the prediction of molecular properties.

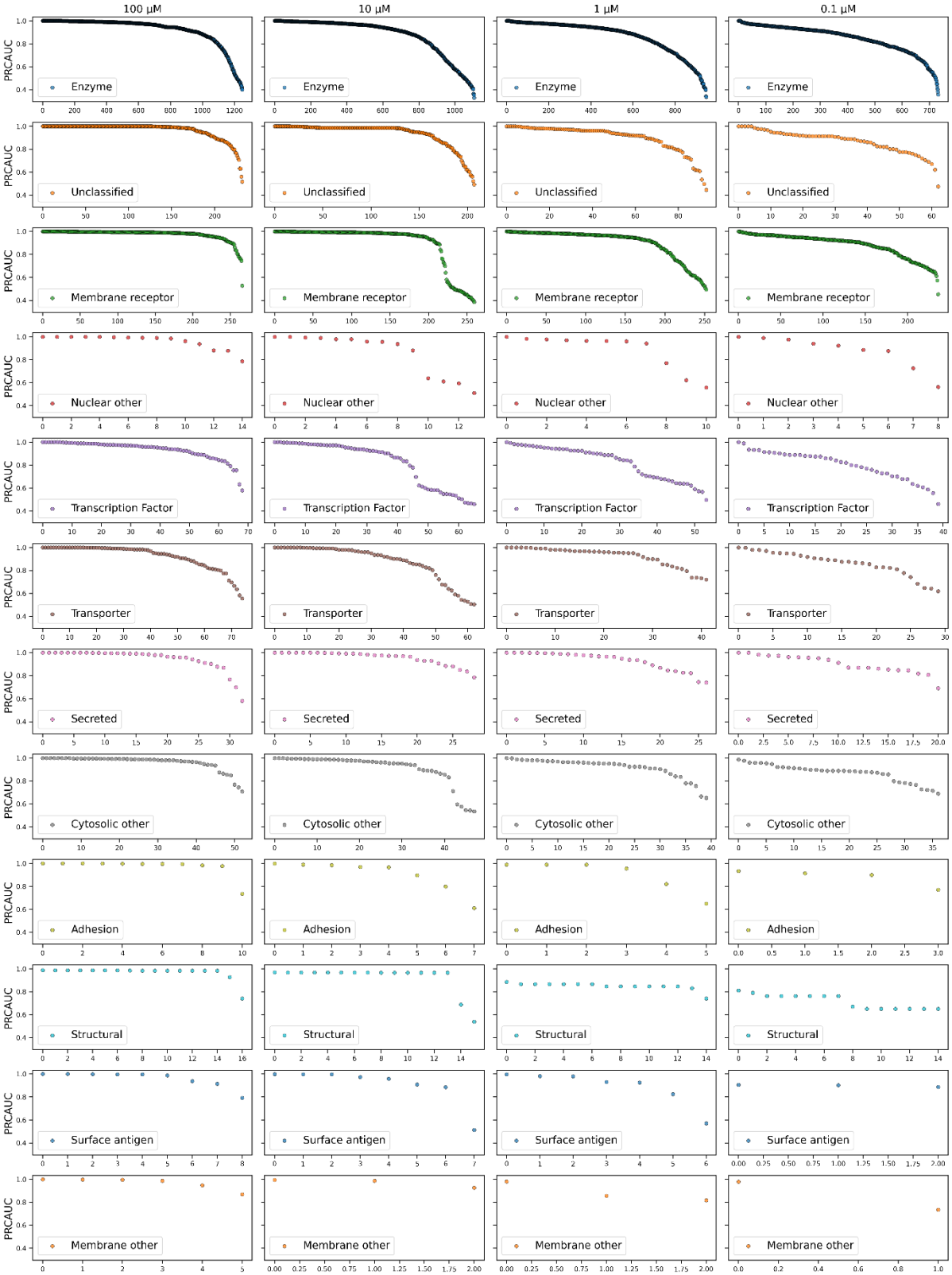


Figure 2.3: PIDGINv5 model performance based on protein classification and concentration. Average area under the precision-recall curve of PIDGINv5 model based on 5-fold stratified scaffold split, categorized by the concentration determining active/inactive cut-off during training (columns) and by ChEMBL target classification (rows).

Ligand preparation

Several ligand preparation protocols are available as prerequisites to docking scoring functions ensuring that molecules undergo appropriate stereoisomer enumeration, tautomer enumeration and protonation. Unless otherwise specified, all ligand preparation protocols are parallelizable using Dask[206], such that multiple processes can be shared across a compute cluster.

Gypsum-DL[217] is an open-source, free to use ligand preparation protocol making use of RDKit to conduct desalting, stereoisomer and tautomer enumeration, and 3D embedding. Meanwhile, Dimorphite-DL[227] is used to conduct protonation at a user-specified pH. Here I use Gypsum-DL's multiprocessing parallelization protocol instead of Dask.

LigPrep[218] is part of the licensed Schrodinger software suite which must be installed on the operating system before hand with an appropriate license. LigPrep conducts molecule desalting, stereoisomer enumeration, tautomer enumeration, and 3D embedding. Meanwhile, LigPrep uses Epik[219] to conduct protonation at a user-specified pH and pH tolerance.

Epik is a streamlined protocol that uses RDKit for stereoisomer enumeration and 3D embedding, bypasses LigPrep, and uses Epik directly for protonation. This enables the specification of only returning the most prominent protonatable state (as opposed to returning all possible protonation states as done by LigPrep). As Epik is part of the licensed Schrodinger software suite, this must be installed on the operating system before hand with an appropriate license.

Moka is a protocol that uses the MoKa software[220] (part of the licensed Molecular Discovery software suite) for ligand protonation returning molecules with protonation states with an abundance above 20% at a pH of 7.4. Following protonation, Corina[228] is used for embedding molecules into 3D space, adding implicit hydrogens and enumerating unspecified stereoisomers. As MoKa and Corina are licensed softwares, they must first be installed on the operating system with appropriate licenses.

Docking

Molscore contains interfaces to several docking software available for users. Protein preparation is not handled automatically and is recommended to be conducted beforehand. In each case, the minimum docking score of any prepared molecule

variant (see ligand preparation above) is returned as the docking score. Unless otherwise specified, all docking protocols are parallelizable using Dask, such that multiple processes can be shared across a compute cluster.

Glide[212] is part of the licensed Schrodinger software suite which must be installed on the operating system before hand with an appropriate license. To run Glide, a template input file (this can be generated by configuring Glide in Maestro and then specifying 'Write' *via* dropdown options writing a file with the suffix '.in') is required which specifies the path of the docking grid and any additional docking constraints to be run (any specified path to existing ligand files in the input file will be ignored). Therefore, a user must first generate a docking grid.

PLANTS[213] is a licensed docking software (free for Academics) which must be installed on the operating system before hand with an appropriate license. To run PLANTS, a receptor file and reference ligand file (to automatically identify the docking box) is required.

GOLD[214] is a licensed docking software that is part of the CCDC software suite which must be installed on the operating system before hand with an appropriate license. To run GOLD, a receptor file and reference ligand file (to automatically identify the docking box) is required. A default configuration file is used specifying docking parameters; however, a user-specified configuration file can be provided.

OEDock is a licensed docking software that is part of the OpenEye software suite which is installed as a pre-requisite specified in the MolScore environment; however, an appropriate license is required. To run OEDock, a receptor file and reference ligand file is required. Conformations are generated by Omega and then either FRED[229] or Hybrid[230] docking algorithms can be utilised.

Smina[216] is a free open-source software that is installed as a pre-requisite specified in the MolScore environment. To run Smina, a receptor file and reference ligand file (to automatically identify the docking box) is required.

Synthesizability

SAscore[172] is a measure of synthesizability based on fragment presence in known molecules and an estimation of molecular complexity. It is available in the molecular descriptors scoring function as described previously.

RAscore[179] is a predictive model trained to predict the outcome of the AiZynthFinder[201] computer-aided synthesis planning software i.e., whether a synthetic route solution can be proposed. Therefore, molecules are correspondingly scored based on their predicted probability of there being an AiZynthFinder solution. The pre-trained models shared by Thakkar et al. are available for the user to choose from (i.e., ChEMBL, GDB or GDBMedChem using either XGBoost or a deep neural network). Due to library incompatibilities and the specific version of XGBoost[231] required, this scoring function is run as a subprocess within the corresponding conda environment proposed by the authors. Molscore will automatically look for the correctly named conda environment and if not installed, attempt to install it.

AiZynthFinder[201] is an open-source computer-aided synthesis planning model based on seminal work by Segler et al.[232]. This model attempts to identify synthetic route proposals for molecules and therefore molecules can be scored based on whether a route is solved (binary score), the top score (MCTS reward), the number of steps or the number of precursors. The original policies, templates and stocks shared by the authors are available, or user-specified policies, templates and stocks can be specified. Note that relative to other scoring functions available this is computationally expensive to run. Due library incompatibilities, this scoring function is run as a subprocess within the corresponding conda environment proposed by the authors. Molscore will automatically look for the correctly named conda environment and if not installed, attempt to install it.

2.2.1.2 Transformation functions

Transformation functions can be used to transform a molecule's parameter (x_i) returned from a scoring function to a value between zero and one ($x_i' \in (0, 1)$), visual examples of the transformation functions are shown in Figure 2.4.

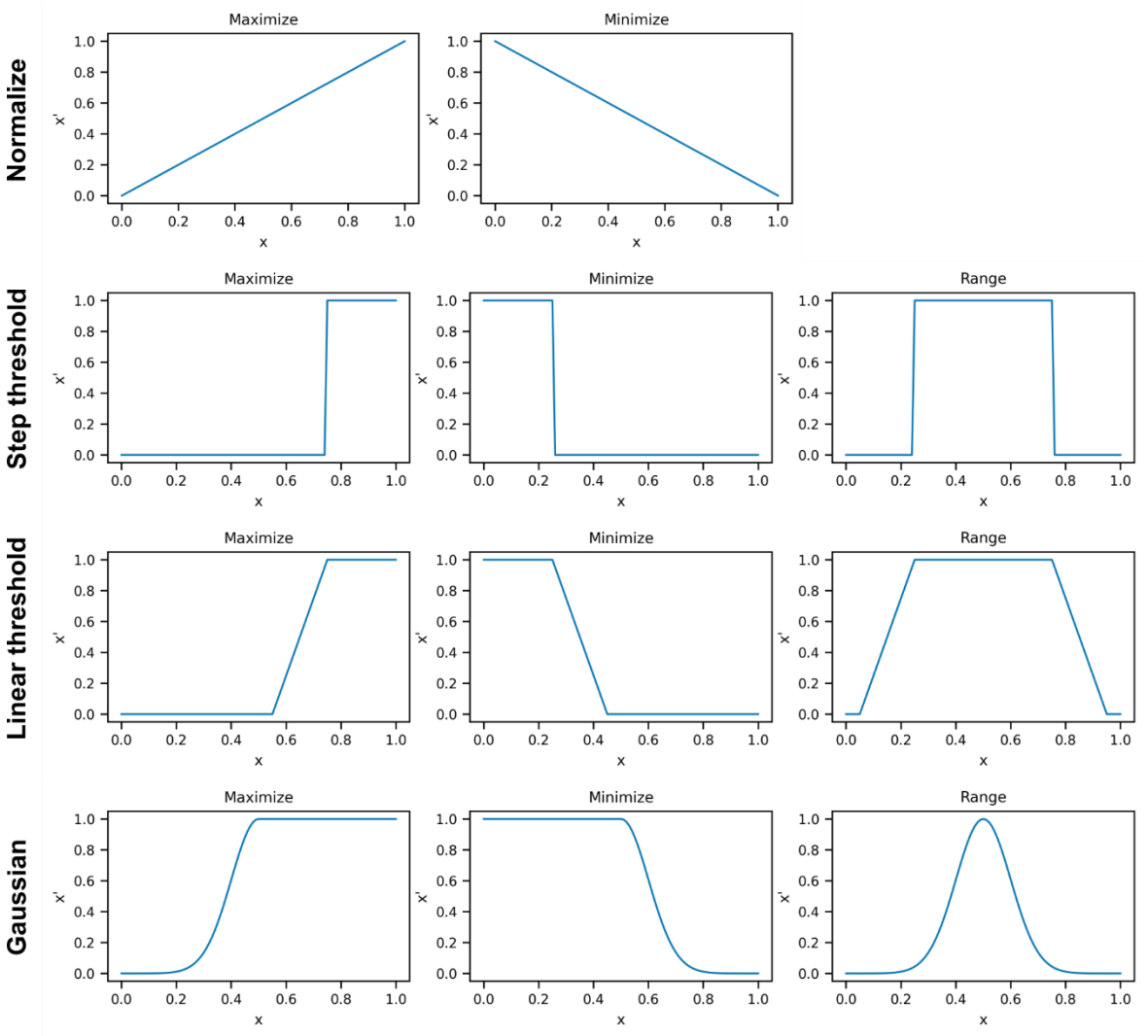


Figure 2.4: Transformation function examples applied to returned parameters to map into the range zero to one.

Normalize applies max-min normalization $f_{maxmin}(x; x_{max}, x_{min})$ to a respective parameter based on either the specified or observed maximum and minimum values for that parameter (if not specified, maximum and minimum values are updated during the course of optimization). If the objective is to minimize the respective parameter, then the maximum and minimum values are switched in the equation below.

$$x_i' = \begin{cases} \dfrac{x_i - x_{min}}{x_{max} - x_{min}}, & if\ maximize \\ \dfrac{x_i - x_{max}}{x_{min} - x_{max}}, & if\ minimize \end{cases}$$

Equation 2.1

Step threshold applies a step transformation to a respective parameter transforming it to either zero or one based on a specified threshold (t), or if the

objective is to obtain a value in a specific range, then two thresholds are used (t_{upper}, t_{lower}).

$$x_i' = \begin{cases} \begin{cases} 1, & if\, x_i \geq t \\ 0, & else \end{cases} & if\ maximize \\ \begin{cases} 1, & if\, x_i \leq t \\ 0, & else \end{cases} & if\ minimize \\ \begin{cases} 1, & if\ t_{lower} \leq x_i \leq t_{upper} \\ 0, & else \end{cases} & if\ range \end{cases}$$

Equation 2.2

Linear threshold[54] applies max-min normalization to a respective parameter if it is above/below a threshold (t) plus/minus a buffer (b) depending on whether to the objective is to maximize or minimize the parameter. If the objective is to obtain a value in a specific range, then two thresholds are used (t_{upper}, t_{lower}).

$$x_i' = \begin{cases} \begin{cases} 1, & if\ x_i \geq t \\ 0, & if\ x_i \leq t - b \\ f_{maxmin}(x_i; t, t-b), & if\ t - b < x_i < t \end{cases} & if\ maximize \\ \begin{cases} 1, & if\ x_i \leq t \\ 0, & if\ x_i \geq t + b \\ f_{maxmin}(x_i; t, t+b), & if\ t + b > x_i > t \end{cases} & if\ minimize \\ \begin{cases} 1, & if\ t_{lower} \leq x_i \leq t_{upper} \\ 0, & if\ x_i \leq t_{lower} - b\ or\ x_i \geq t_{upper} + b \\ f_{maxmin}(x_i; t_{lower}, t_{lower} - b), & if\ t_{lower} - b < x_i < t_{lower} \\ f_{maxmin}(x_i; t_{upper}, t_{upper} + b), & if\ t_{upper} + b > x_i > t_{upper} \end{cases} & if\ range \end{cases}$$

Equation 2.3

Gaussian threshold[54] applies a Gaussian transformation to a respective parameter based on a specified mean (μ) and sigma (σ) value and can be used to maximize, minimize, or achieve a certain range for a specified parameter.

$$x_i' = \begin{cases} \begin{cases} 1, & if\ x_i \geq \mu \\ e^{\frac{(\frac{x_i - \mu}{\sigma})^2}{2}}, & else \end{cases} & if\ maximize \\ \begin{cases} 1, & if\ x_i \leq \mu \\ e^{\frac{(\frac{x_i - \mu}{\sigma})^2}{2}}, & else \end{cases} & if\ minimize \\ e^{\frac{(\frac{x_i - \mu}{\sigma})^2}{2}}, & if\ range \end{cases}$$

Equation 2.4

2.2.1.3 Aggregation functions

Aggregation functions define how multiple parameters (x_i) for a respective molecule are combined into a final score ($S(m)$) or reward in range zero to one (($S(m) \in (0,1)$).

Weighted sum combines n parameters for a molecule by assigning a weight (w_i) to each parameter (x_i) and then summing the weighted parameters. Weights are normalized by the total number of parameters such that any positive value can be used as a weight ($w_i \in \mathbb{R}_+$).

$$S(m) = \frac{\sum_i^n x_i w_i}{\sum_i^n w_i}$$

Equation 2.5

Auto-weighted sum[221] combines n parameters for a molecule by automatically assigning a weight (w_i) to each parameter (x_i) and then summing the weighted parameters as above. Weights are automatically assigned based on the fraction of molecules scoring above a specified threshold ($t = 0.5$) within a batch of molecules where the number of molecules with parameter x_i in a batch is defined as $C(x_i)$.

$$w_i = \frac{C(x_i \geq t)}{C(x_i)}$$

Equation 2.6

Product combines n parameters for a molecule by calculating the product of all parameters (x_i). The resulting score is in the range zero to one ($S(m) \in (0,1)$) because each parameter is in the range zero to one ($x_i \in (0,1)$) due to the transformation functions previously applied.

$$S(m) = \prod_i^n x_i$$

Equation 2.7

Weighted product combines n parameters for a molecule by calculating the product of all parameters (x_i) with an assigned weight (w_i) normalized by the sum of the weights such that any positive value can be used as a weight ($w_i \in \mathbb{R}_+$).

$$S(m) = \left[\prod_i^n {x_i}^{w_i}\right]^{\frac{1}{\sum_i^n w_i}}$$

Equation 2.8

Auto-weighted product[221] combines n parameters for a molecule by automatically assigning a weight (w_i) to each parameter (x_i) and then calculating the weighted product as above. Weights are automatically assigned based on the fraction of molecules scoring above a specified threshold ($t = 0.5$) within a batch of molecules – as for the auto-weighted sum.

Geometric mean combines n parameters for a molecule by calculating the square root of the product of parameters (x_i).

$$S(m) = \left[\prod_{i}^{n} x_i\right]^{\frac{1}{n}}$$

Equation 2.9

Arithmetic mean combines n parameters for a molecule by calculating the average value of respective parameters (x_i).

$$S(m) = \frac{1}{n}\sum_{i}^{n} x_i$$

Equation 2.10

Pareto front[221] scores molecules based on their pareto rank per batch of molecules. Molecules are first sorted into pareto fronts from dominated solutions to non-dominated solutions and then ranked based on intra-molecular distance within each pareto front based on Tanimoto distance of ECFP6 (2,048 bits), with most distant ranked first. Molecules are classed as $desirable$ if all parameters (x_i) are above a specified threshold ($t = 0.5$) i.e., $\forall_i : x_i \geq t$ and otherwise $undesirable$. The score ($S(m)$) is then calculated based on a molecules index in the pareto rank (k) and desirability relative to the ratio of $desirable$ and $undesirable$ molecules within the batch. For further detail I refer the reader to Liu et al.[221].

$$S(m) = \begin{cases} 0.5 + \dfrac{k - N_{undesirable}}{2N_{desirable}}, & if\ desirable \\ \dfrac{k}{2N_{undesirable}}, & if\ undesirable \end{cases}$$

Equation 2.11

2.2.1.4 Diversity filters

Diversity filters (DFs) serve as an additional filter to penalize molecules that are generated due to generative model exploitation. In each case, they decrease the molecules final score ($S(m)$) by a differing amount depending on how much the generated molecule is exploitative in nature.

Unique is a DF that transforms a molecule's score to zero if the molecule is non-unique (i.e., has been previously generated by the generative model).

$$DF\big(S(m)\big) = \begin{cases} S(m), & if\ unique \\ 0, & else \end{cases}$$

Equation 2.12

Occurrence is a DF that linearly penalizes the score ($S(m)$) of 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 ($Buff$).

$$DF(S(m)) = \begin{cases} S(m) \times \dfrac{Occ - (Tol + Buff)}{Tol + Buff}, & if\ Tol\ < Occ\ < Buff \\ S(m), & if\ Occ\ \leq Tol \\ 0, & if\ Occ\ \geq Buff \end{cases}$$

Equation 2.13

The following DFs are from or adapted from Blaschke et al.[167]. Each DF creates a memory of generated molecules and clusters them into different bins. As the number of molecules in a bin increase beyond a particular threshold, new molecules belonging to that bin are penalized. Thus, penalizing over-exploited areas of chemical space. Each different DF below defines how the molecules are clustered, and each DF can be modified by the following hyperparameters:

1) *Binsize* – the number of molecules in a bin (i.e., cluster) before penalization starts to occur.
2) *Minscore* – the minimum score before passed to the diversity filter i.e., what threshold to consider molecules for DF penalization.
3) *Outputmode* – how to penalize a molecule's score out of the following three options.

a. Binary – penalize a molecule if the addition of a molecule results in a bin index (k) that exceeds the allowed *binsize*, returning a score of zero.

$$DF\big(S(m)\big) = \begin{cases} 1, & if\ k \leq binsize \\ 0, & else \end{cases}$$

b. Linear – penalize a molecule's score according to the bin index (k) and allowed *binsize via* a linear transformation function.

$$DF\big(S(m)\big) = \begin{cases} 0, & if\ k \geq binsize \\ S(m) \times \dfrac{binsize - k}{binsize}, & else \end{cases}$$

c. Sigmoid – penalize a molecule's score according to the bin index (k) and allowed *binsize via* a sigmoid transformation function.

$$DF\big(S(m)\big) = \begin{cases} 0, & if\ k \geq binsize \\ S(m) \times \left(1 - \dfrac{1}{1 + e^{\frac{-\left(2\left(\frac{k}{binsize}\right)-1\right)}{0.15}}}\right), & else \end{cases}$$

Identical Murcko assigns a molecule to a cluster if it contains the same Bemis-Murcko scaffold as the cluster centroid.

Identical Topological Scaffold assigns a molecule to a cluster if it contains the same generic scaffold i.e., the Bemis-Murcko scaffold but considering every atom as a carbon atom and every bond as a single bond.

Compound Similarity assigns a molecule to a cluster if the Tanimoto similarity based on ECFP4 (2,048 bits) fingerprints is greater than or equal to 0.6 from the cluster centroid.

Scaffold Similarity Atom Pair assigns a molecule to a cluster if the Tanimoto similarity based on Atom Pair fingerprints is greater than or equal to 0.6 from the cluster centroid.

Scaffold Similarity ECFP assigns a molecule to a cluster if the Tanimoto similarity based on ECFP4 (1,024 bits) fingerprints of the corresponding molecule's Bemis-Murcko scaffold is greater than or equal to 0.8 from the Bemis-Murcko scaffold of the cluster centroid.

2.2.2 moleval

The `moleval` sub-package is largely an extension of the MOSES[79] suite of evaluation metrics computed for *de novo* molecules given a set (or sets) of reference molecules. The main element of this sub-package is the `GetMetrics` class found in the `metrics.py` module. This is initialised by optionally specifying some reference datasets (for example, train and test sets used for the measurement of extrinsic properties), and it then takes as input a list of *de novo* molecules and outputs the respective calculated metrics. Additionally, the CSV output file written by `molscore` can be provided to the `statistics_by_n.py` script, which computes evaluation metrics and basic statistics (mean, median and standard deviation) per *n* molecules or *n* column values (e.g., per 100 steps).

Table 2.3 highlights all the evaluation metrics available in `moleval` split into intrinsic properties (based solely on *de novo* molecules) and extrinsic properties (in reference to an external dataset). Some additional metrics not found in MOSES for intrinsic properties include sphere exclusion diversity (SEDiv)[233], scaffold uniqueness, scaffold diversity, functional group and ring system diversity[104] and a measure of purchasability in the ZINC20 in-stock catalogue using molbloom[234,235]. Additional metrics for extrinsic properties include analogue similarity[167] and coverage, functional group and ring system similarity[104] and average fraction of outlier bits (a.k.a. ‘Silliness’[236]) i.e., the average ratio of ECFP4 fingerprint bits not found in the reference dataset indicating.

Table 2.3: Evaluation metrics available in the `moleval` sub-package. No metrics require a license.

		References
Intrinsic properties	Validity	54,79
	Uniqueness	54,79
	Scaffold uniqueness	
	Internal diversity (1 & 2)	79,162
	Sphere exclusion diversity	147
	Scaffold diversity	
	Functional group diversity	104
	Ring system diversity	104
	Filters (MCF & PAINS)	79
	Purchasability	234
Extrinsic properties	Novelty	54,79
	FCD	165
	Analogue similarity	167
	Analogue coverage	
	Functional group similarity	
	Ring system similarity	
	Single nearest neighbour similarity	79
	Fragment similarity	79
	Scaffold similarity	79
	Outlier bits (Silliness)	236
	Wasserstein distance (LogP, SA Score, NP score, QED, Weight)	79

2.2.2.1 Performance metrics

The following performance metrics are computed on generated *de novo* molecules, where G is a formal set of valid, unique molecules. Extrinsic properties are calculated in reference to a reference set of molecules R. Unless otherwise specified, RDKit was used for implementation of molecular operations. All commonly implemented metrics in Table 1.1 and Table 1.2 are included with the exception of GuacaMol Filter, functional group metrics, and ring system metrics (which are adapted to similarity metric). New or adapted metrics are described below.

Sphere exclusion diversity (SEDiv@1k) is a measure of the datasets diversity as approximated by the fraction of molecules required to explain the chemical space. More concretely, the sphere exclusion algorithm is applied to cluster molecules according to a distance cut-off of an ECFP4 (1,024 bit) Tanimoto distance of 0.65 (where any more similar approximately corresponds to 80% probability of possessing similar bioactivity). This ensures that no two cluster centroids are more similar than the specified cut-off and therefore represent a set of diverse molecules G_d. To allow comparison between different sets of different size, this metric should be run on a random sample of 1,000 molecules from a generated set as a representative sample where $g \subset G \; and \; |g| = 1000$.

$$SEDiv@1k(G) = \frac{|g_d|}{|g|}$$

Equation 2.14

Scaffold diversity (ScaffDiv) is the same as internal diversity (see Table 1.1) instead applied to the Bemis-Murcko scaffolds of molecules instead.

Scaffold uniqueness (ScaffUniqueness) is the same Uniqueness (see Table 1.1) except applied to the Bemis-Murcko scaffolds of molecules instead.

Functional group diversity (FG) is the fraction of unique functionality groups fg compared to all functional groups present in the set of generated molecules. Where functional groups are identified by as described by Ertl et al.[168] and implemented in RDKit, also used in Zhang et al.[104]. Note that functional groups include both functional group atoms and their immediate connect unmarked carbon environment.

$$FG(G) = \frac{|G_{fg}|}{|G^*_{fg}|}$$

Equation 2.15

Ring system diversity (RS) is the fraction of unique ring systems rs compared to all ring systems present in the set of generated molecules. Where ring systems are identified using RDKit, also used in Zhang et al.[104].

$$RS(G) = \frac{|G_{rs}|}{|G^*_{rs}|}$$

Equation 2.16

Purchasability (PurchasabilityZINC20) is the estimated fraction of molecules found contained in the ZINC20[235] in-stock catalogue as identified via molbloom[234].

Analogue coverage (AnCov) is the fraction of the reference set of molecules with an identified analogue contained in the generated set. An analogue is defined as with analogue similarity.

$$AnCov(G,R) = \frac{|\{m_r \in R | \exists m_G \in G(sim(m_g, m_r) \geq 0.4)\}|}{|R|}$$

Equation 2.17

Functional group similarity (FG) is the cosine similarity (implemented by scipy[237]) between the identically ordered count vectors of the functional groups (as described previously) in the generated set ($\vec{G}_{fg}$) and reference set ($\vec{R}_{fg}$).

$$FG(G,R) = \frac{\vec{G}_{fg} \cdot \vec{R}_{fg}}{||\vec{G}_{fg}||\ ||\vec{R}_{fg}||}$$

Equation 2.18

Ring system similarity (RS) is the cosine similarity (implemented by scipy) between the identically ordered count vectors of the ring systems (as described previously) in the generated set ($\vec{G}_{rs}$) and reference set ($\vec{R}_{rs}$).

$$RS(G,R) = \frac{\vec{G}_{rs} \cdot \vec{R}_{rs}}{||\vec{G}_{rs}||\ ||\vec{R}_{rs}||}$$

Equation 2.19

Outlier Bits (a.k.a Silliness) is the average fraction of ECFP4 bits in generated molecules not contained anywhere in a complete set of all reference dataset fingerprint bits R_{bits}.

$$OutlierBits(G,R) = \frac{1}{|G|}\sum_{i \in G}\frac{\#ECFP4(m_i) \notin R_{bits}}{\#ECFP4(m_j)}$$

Equation 2.20

2.2.3 Implementation challenges

A particular challenge when combining a variety of scoring functions and software from published methods is conflicting library dependencies. In addition, predictive models should use the same version of a respective library (e.g., Scikit-Learn) during prediction as was used during training where possible, as there may be subtle changes to the source code affecting the prediction. In cases where published methods require specific library versions that conflict with the `molscore` Python environment or must be consistent with those used during training (currently, AiZynthFinder[201] and RAscore[179]), scoring function scripts are run as a subprocess from their respective conda environment with the dependencies as specified by the authors. In order to improve user experience, `molscore` will check for these separate conda environments and if not present, attempt to create them automatically when the scoring function is used for the first time. However, there are several caveats to this approach, the first being the assumption of the use of conda for environment management, and the second being computational performance decrease. The performance decrease is because running a scoring function *via* a subprocess requires redundant loading of the relevant conda environment, redundant loading of scoring function parameters and I/O operations to send SMILES and receive respective scores via writing and reading to disk, each time. This sub-optimal process adds to the overall wall time when using `molscore` which is negligible in the case of long running scoring functions like AiZynthFinder[201], but sub-optimal when using fast running scoring functions like RAscore[179]. In future versions of `molscore`, I will move to running methods with conflicting/specific library dependencies as local servers to avoid redundant operations and send/receive data in memory instead, which will improve computational performance and overall software stability.

2.2.4 Implementing custom user scoring functions

It is sometimes desirable for a user to implement their own scoring function. The simple, modular design enables easy implementation of a user-defined scoring functions in Python. The user must create a new module in the `scoring_functions`

directory and write a class for their function. The class must adhere to the following 4 requirements: 1) The class should have a class attribute `return_metrics` listing the name of metrics returned. 2) The class must take a `prefix` parameter argument in `__init__` which is inserted before any return metrics separated by an underscore, for example, ‘prefix_score1’. 3) The class `__call__` method must accept a list of SMILES and return their respective scores as a list of dictionaries in the same order. 4) If a SMILES results in an invalid molecule, the score returned should be 0. Finally, the class must be added to the `scoring_functions/__init__.py` to enable its use. Note that if PyCharm style documentation and python typing is used, the user implemented scoring functions should automatically appear with correct descriptions and widgets in the configuration GUI.

2.3 Results and discussion

2.3.1 User interface

After installation, MolScore can be implemented into a goal-directed generative model optimization scheme in just three lines of code. All that is needed as input is the name of the model, the configuration file that describes the objective (i.e., how to score molecules) and SMILES that require scoring.

2.3.1.1 Generative model integration

Three lines of python code are needed to integrate MolScore into an existing generative model scheme:

```
1. from molscore.managar import MolScore
2. ms = MolScore(model_name=’my_model’,
   task_config=’my_task.json’)
3. ms.score(SMILES)
```

Where SMILES is a list of SMILES, and where the last line is repeated as many times as necessary. An explicit step number can be provided during scoring, if not, it will iteratively count up from one.

2.3.1.2 Writing a configuration file

The configuration file that describes the objective and how to score molecules is a JSON formatted file read and parsed by the `MolScore` class (example shown in Figure 2.5a). This consists of sections to specify logging parameters, scoring functions to run,

diversity filters to use, and scoring function metrics and how to aggregate them to score the final molecule. However, writing these manually configuration files manually is tedious and requires documentation of every possible option to specify which must be read and interpreted by the user without any syntax mistakes or typos. Therefore, a Streamlit app is provided to easily write configuration files interactively with descriptions (see Figure 2.5b). The app can be run via the `streamlit run molscore/gui/config.py` command that loads the GUI in a web browser. This facilitates configuration writing and automatically parses the options specified into the correctly formatted JSON configuration file. Moreover, docstrings and typing are interpreted to provide descriptions and automatic widgets, such that if a user implements a custom scoring function (as described in 2.2.4), it will be automatically parsed and available to specify in the GUI.

a)


```
{
 "task": "Albuterol_similarity",
 "output_dir": "./",
 "load_from_previous": false,
 "logging": false,
 "monitor_app": false,
 "diversity_filter": {
  "run": false
 },
 "scoring_functions": [
 {
  "name": "TanimotoSimilarity",
  "run": true,
  "parameters": {
   "prefix": "Albuterol",
   "ref_smiles": [
    "CC(C)(C)NCC(O)c1ccc(O)c(CO)c1"
   ],
   "radius": 2,
   "bits": 1024,
   "features": true,
   "counts": true,
   "method": "max",
   "n_jobs": 1
  }
 }
 ],
 "scoring": {
  "method": "single",
  "metrics": [
   {
    "name": "Albuterol_Tc",
    "weight": 1.0,
    "modifier": "lin_thresh",
    "parameters": {
     "objective": "maximize",
     "upper": 0.75,
     "lower": 0.0,
     "buffer": 0.75
    }
   }
  ]
 }
}
```


b)

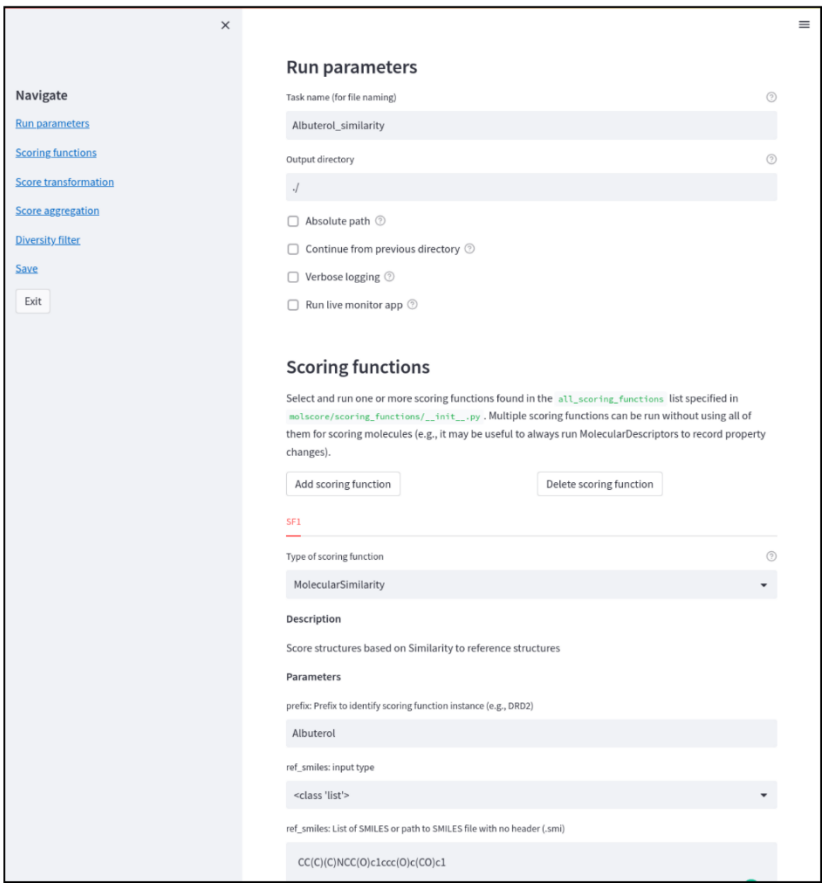


Figure 2.5: Molscore input GUI. (a) Example configuration file reimplementing the Albuterol Similarity GuacaMol task. (b) Streamlit app to aid the creation of new configuration files and avoid manual writing of JSON files. The app annotates options available to the user and automatically parses it into the required JSON format.

2.3.1.3 Monitoring and analysing de novo molecules

A Streamlit app to monitor *de novo* molecule generation in real-time or analyse results post-hoc is also provided (see Figure 2.6). This is useful to gain quick insights into generative model behaviour with respect to chemistry generated, without needing to wait until the end of optimization (especially in the case of computationally expensive scoring functions). This is run automatically during optimization if specified in the configuration file, alternatively, it can be run manually at any time via the `streamlit run molscore/gui/monitor.py` command. The app loads a graphical user interface in a web browser and contains functionality to check any variable scored including validity and uniqueness, select and visualise 2D molecular graphs, assess clusters identified by an appropriate diversity filter (if used), and export selected or top *k* molecules. In addition, if a scoring function is used that results in 3D coordinate files and PyMol[238] is installed, PyMol will be loaded and selected molecules can be exported directly into PyMol. Lastly, other pre-existing `molscore` *de novo* molecule generation results can be loaded for quick comparison between runs.

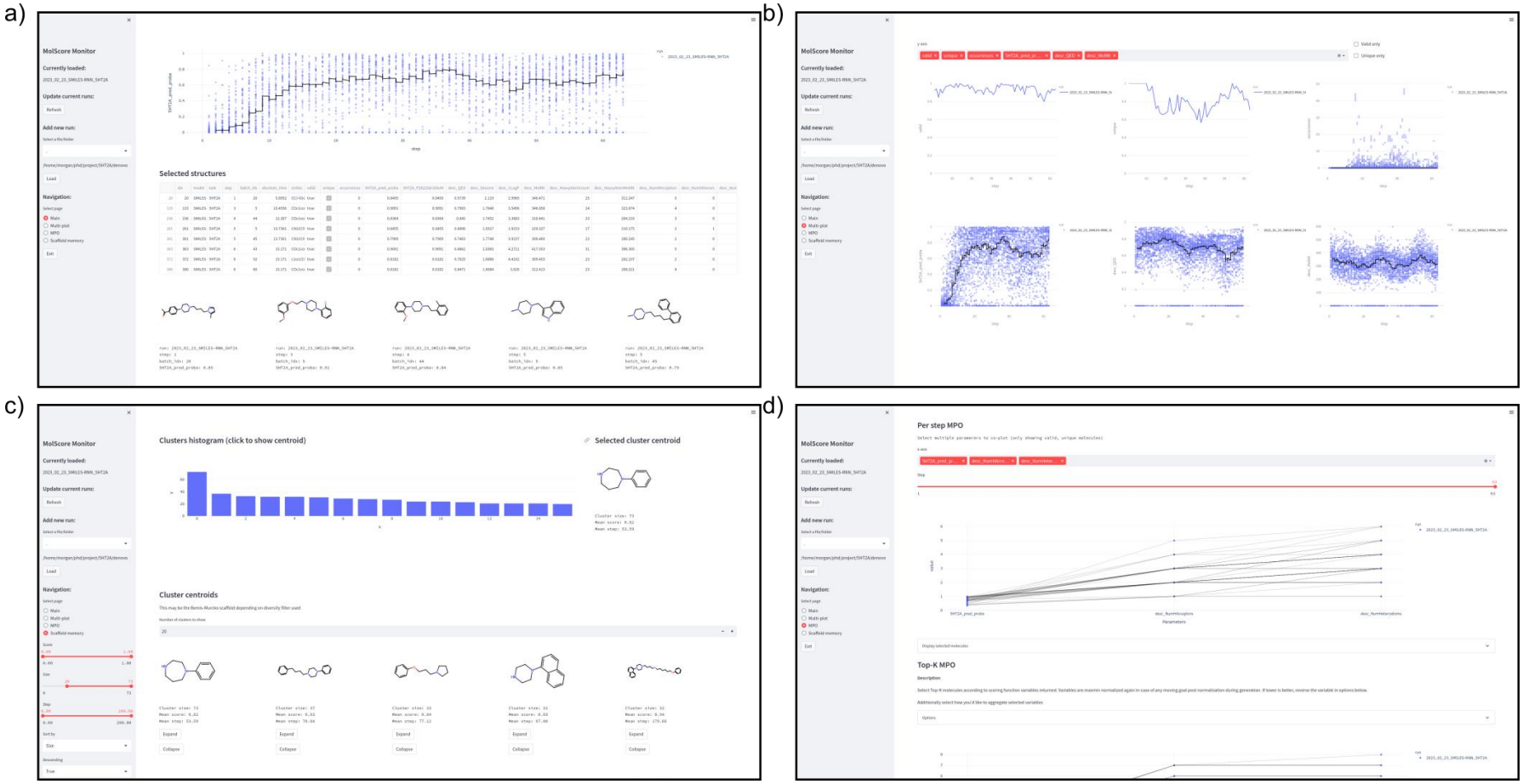


Figure 2.6: Molscore Streamlit app that can be run during or after goal-directed generative model optimization. Here showing optimization of 5HT2A predicted probability of activity. (a) Main page used to plot training progress and select, visualize, and export molecules. (b) Multi-plot page to visualize many variables at the same time. (c) Scaffold memory analysis (if using an appropriate diversity filter) to visualize the clusters and respective scaffolds of chemotypes generated. (d) Parallel plots to assess multi-parameter optimal compounds, as well as the top *k* overall compound.

2.3.2 Sphere exclusion to measure chemical diversity

As suggested in 1.2.3.1, the internal diversity (IntDiv), a common metric of chemical diversity, can be difficult to interpret due to the double average losing the notion of the underlying distribution, as well as the well-known confounding effect of heavy atom count on Tanimoto similarity[174]. Therefore, I propose a new metric to measure the diversity of *de novo* compounds called sphere exclusion diversity (SEDiv). SEDiv is the fraction of diverse compounds selected using the sphere exclusion algorithm[163] at a set Tanimoto threshold, as described in 1.1.1.1. Note that there is no underlying ground truth for the diversity of chemical dataset and so a metric should align with empirical observations, intuition and be useful in practice.

To investigate differences between IntDiv and SEDiv further, I subset ChEMBL28[239] to only include molecules with 5-50 heavy atoms and randomly sampled 500 molecules either side of a heavy atom threshold, for thresholds 10-45 in increments of 1 (with 10 repeats per threshold) – to mimic datasets biased towards smaller or larger molecules approximated by the mean heavy atom count in each sample. It is interesting to note the drop off in the count of molecules with above ~33 heavy atoms, where chemistry likely drifts into more undesirable chemical space for molecules with drug-like properties (the primary focus of ChEMBL), and so less drug discovery research and data depositions are made here. There is a clear decrease in IntDiv with an increase in the mean number of heavy atoms in accordance with the hypothesized confounding effect[174] (see Figure 2.7b). On the other hand, SEDiv (Figure 2.7c) shows a similar trend to the count of molecules per heavy atom bin (Figure 2.7a), where a higher molecule count should correspond to a higher likelihood of random samples being more diverse. Thus, the trend observed for SEDiv with heavy atom count aligns better with intuition.

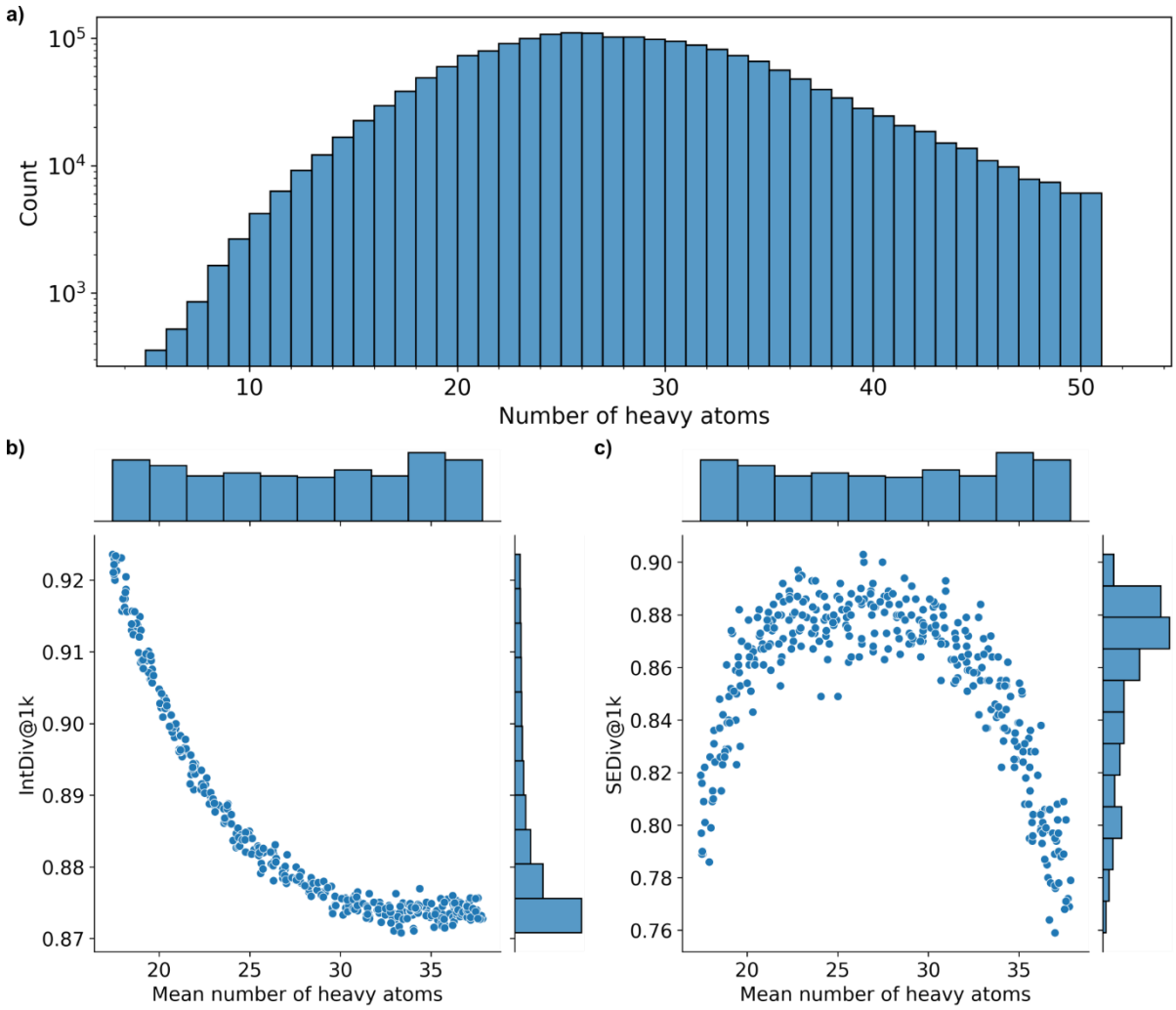


Figure 2.7: Investigation of the dependence of internal diversity (IntDiv) on molecular size. Shown here is (a) the count of molecules in ChEMBL28 from 5-50 heavy atoms, (b) the relationship between the mean number of heavy atoms in different samples and their IntDiv@1k and (c) the relationship between the mean number of heavy atoms in different samples and their SEDiv@1k. Where "@1k" refers to a sample size of 1,000. Note that the range of heavy atoms in (b) and (c) is smaller than in (a) as it reflects the mean of the sample. Additionally, the distribution above (b) and (c) are uniform indicating uniform sampling of distributions with different numbers of mean heavy atoms.

To investigate the difference between SEDiv and IntDiv further, I calculate these two metrics on random subsets of different libraries (Figure 2.8): enumerated virtual libraries of stable molecules up to 17 and 13 heavy atoms (GDB17[49], GDB13[32]), characterised molecules with varying bioactivities (ChEMBL28[239]), a synthetically accessible diversity orientated virtual library (Enamine diverse[240]), synthetically accessible targeted virtual libraries (Enamine GPCR and Enamine Kinase[241]) and characterised molecules with activity (pChEMBL ≥ 5) against specific target classes (ChEMBL28 Family A GPCRs and ChEMBL28 Kinases) and single targets (ChEMBL28 HERG, ChEMBL28 EGFR and ChEMBL28 DRD2). All datasets were similarly processed to neutralize molecules and retain only those with a molecular

weight less than 500 Da, to ensure a similar ‘drug-like’ chemical space. Most notably, IntDiv measures GDB13 as more diverse than GDB17 – which contradicts chemical intuition, but further confers with hypothesized confounding effects[174]. Furthermore, IntDiv measures molecules active against hERG – a promiscuous target related to cardiotoxicity[242] – as diverse as all molecules reported active against any kinases, any family A GPCRs and more diverse than a virtual library designed for diversity. Conversely, SEDiv measures GDB17 as more diverse than GDB13 (which is better distinguished at larger sample sizes, see Figure A.2) and hERG active molecules as more diverse than single targets (EGFR and DRD2) but not as diverse as all molecules active against any family A GPCR or kinase. Therefore, the proposed approach better aligns with chemical intuition regarding the chemical diversity of known libraries. Furthermore, this approach yields values in the full range of possible values 0-1 (unlike internal diversity which mostly lie in a range of ~0.7-0.9). SEDiv further has a direct interpretation as the fraction required to explain the chemical space; therefore, a comparative reference is not always necessary (unlike IntDiv). However, the values measured here provide some context for sample sizes of 1,000 random molecules, which I recommend for future use in comparing *de novo* molecule diversity.

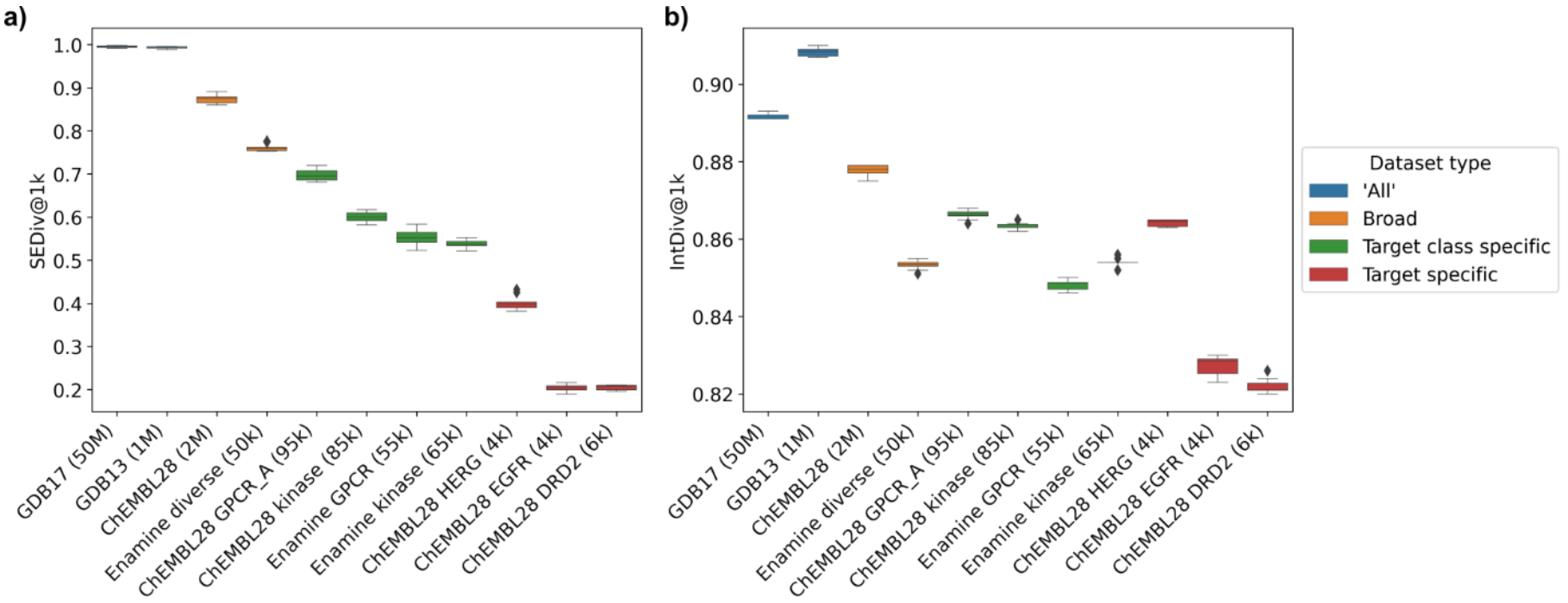


Figure 2.8: Measured diversity of different compound datasets. The measured SEDiv (a) and IntDiv (b) of a randomly sampled 1,000 (@1k) subset of a variety of virtual libraries and datasets of characterised molecules with activity against particular targets belonging to a target class, or single targets. IntDiv shows counterintuitive behaviour such as, measuring GDB13 as more diverse than GDB17 and hERG active molecules as diverse as molecules active against any family A GPCR, any kinase or a virtual library designed towards achieving diversity. Conversely, SEDiv measures diversity in line with chemical intuition.

2.3.3 Molscore use case: Designing 5-HT$_{2a}$ ligands

Here I demonstrate the application of molscore for the design of different, drug discovery relevant objectives, with a focus on the generation of *de novo* 5-HT$_{2a}$ (a.k.a. 5HT2A) ligands as a case study. This is a drug discovery relevant therapeutic target indicated in psychosis and substance-abuse with numerous antagonistic drugs marketed for their use as atypical antipsychotics – with the most recent being Lumateperone[243] approved in 2019 by the FDA. For the purpose of this demonstration, I use a SMILES-based recurrent neural network generative model trained on ChEMBL compounds in combination with Augmented Hill-Climb[244] (introduced in Chapter 4) for molecular optimization.

To start, with I use the functionality available in molscore to design the following first set of objectives:

5HT2A – I use a pre-trained random forest classification model with the PIDGINv5 scoring function to score molecules by their predicted probability of activity at a 1 µM concentration by supplying the 5HT2A uniprot accession.

5HT2A + Synth – To include a measure of synthesizability which is practically needed in any real-world drug discovery campaign, I additionally score molecules by running the RAscore[179] pre-trained models and compute the arithmetic mean of this score together with the predicted probability of 5HT2A activity as before.

5HT2A + BBB – Due to the therapeutic targets prevalence and disease relevance in the central nervous system, I run molecular descriptors and specify certain property ranges that increase the probably of blood brain barrier (BBB) permeability. The property ranges were influenced by Pajouhesh et al.[245]: topological polar surface area below 70, number of hydrogen bond donors below 2, logP between 2 and 4, and molecular weight below 400 Da. Each molecules property value is transformed into the range 0-1 (see Figure A.1) and combined by arithmetic mean with the predicted probability of 5HT2A activity as before.

5HT2A + BBB + Synth – This a combination of all three of the above objectives by arithmetic mean.

Each objective was optimized by the generative model in combination with a diversity filter to penalize exploitation. As shown in Figure 2.9, each of these objectives can be improved during generative model optimization. For reference, 3,771 real compounds with bioactivity values against 5HT2A were extracted from ChEMBL31[204] and their

respective scores based on the first set of objectives are also shown. Surprisingly, the most difficult objective appears to be simple optimization of the 5HT2A predicted probability of optimization; however, I suspect this is largely due to the effect of the diversity filter more heavily penalizing similar molecules for this relatively ‘easy’ task. This is corroborated by running the objective without a diversity filter (see Figure 2.10) which results in quick maximization of this objective, but exploitative mode collapse shortly following (which the use of a diversity filter circumvents). Overall, it appears these objectives are relatively easy to optimize numerically.

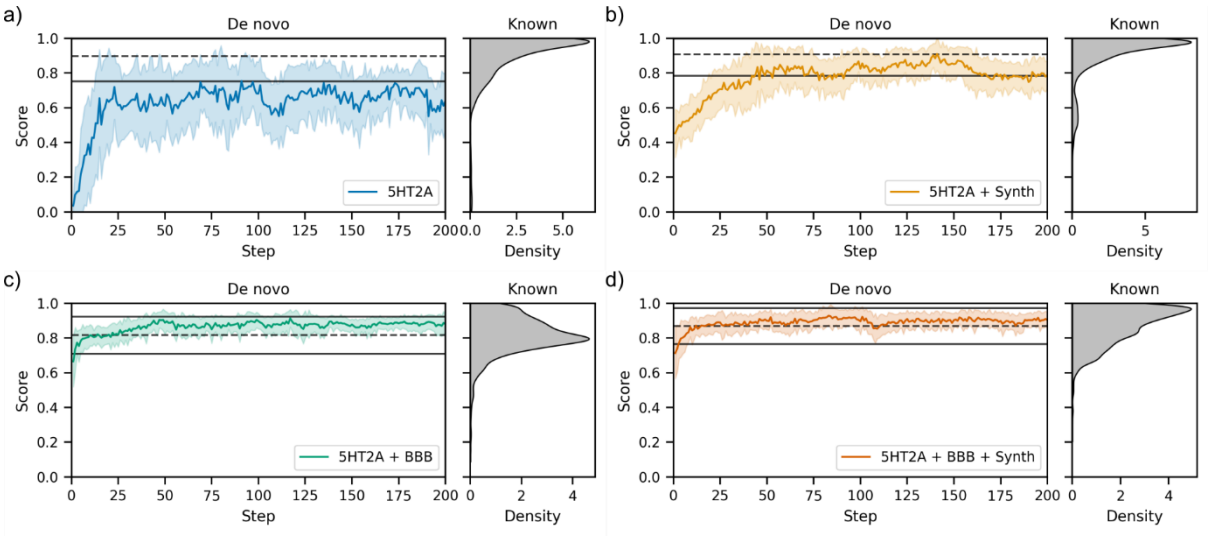


Figure 2.9: *De novo* optimization of the first set of objectives designed by molscore by number of optimization steps (left) with the equivalent score distribution for 3,771 real 5HT2A ligands (right). The dashed line represents the mean of the real ligand distribution and solid lines plus/minus one standard deviation from the mean. (a) The predicted probability of 5HT2A activity at a concentration of 1 µM. (b) The first objective (a) combined with predicted synthesizability by RAscore. (c) The first objective (a) combined with property ranges increasing the probability of BBB. (d) All three objectives (a-c) combined.

.

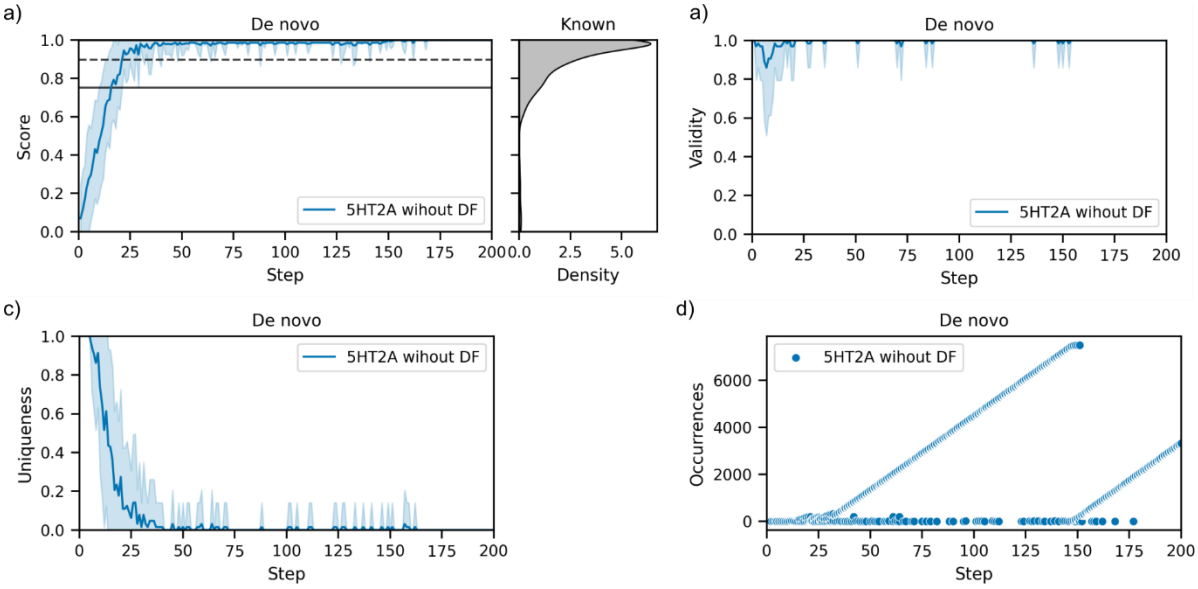


Figure 2.10: *De novo* optimization of 5HT2A without any diversity filter. (a) Optimization of the 5HT2A predicted probability objective without running any diversity filters (DF), as well as (b) validity (c) uniqueness and (d) number of unique occurrences during optimization. Maximal optimization of predicted probability score is achieved quickly compared to known ligands and easily above the range of known 5HT2A ligands. However, a sharp drop in uniqueness is observed signalling mode collapse of the generative model without the use of a diversity filter to penalize exploitation. By looking at the number of occurrences (d) it can be seen that the generative model collapses into generating predominantly just two structures.

As with many drug discovery campaigns, a key challenge for 5HT2A ligands is minimizing off-target bioactivity and achieving pharmacological selectivity. In this case, particularly against dopaminergic receptors (especially DRD2 bound by typical antipsychotics) which leads to extrapyramidal symptoms as serious side-effects[246,247]. As a proxy for desirable selectivity profiles, I design a second set of objectives with molscore particularly utilising PIDGINv5 functionality (as with the first set of objectives a diversity filter is also used):

5HT2A – As before, I use a pre-trained random forest classification model from PIDGINv5 to score molecules by their predicted probability of activity at a 1 µM concentration i.e., no selectivity proxy is used.

5HT2A + Membrane Selectivity – As a proxy for a generic off-target assay, a random forest classification model at a 10 µM concentration for every Class A GPCR targets with sufficient bioactivity data in ChEMBL31 is run (266 out of a possible 312). The prediction is classified into active or inactive (as opposed to taking the predicted probability) for each receptor and the ratio of active predictions is returned as the score. This ratio is transformed so that low ratios have a high score, therefore minimizing this parameter. The arithmetic mean is taken in combination with the predicted probability of activity against 5HT2A.

5HT2A + DRD2 Selectivity – The predicted probability of DRD2 bioactivity at a concentration of 10 µM is minimized in addition to maximizing the predicted probability of activity against 5HT2A.

5HT2A + Dopamine Selectivity – The average predicted probability of bioactivity against each dopaminergic target at a concentration of 10 µM is minimized in addition to maximizing the predicted probability of activity against 5HT2A.

5HT2A + Serotonin Selectivity – The average predicted probability of bioactivity against each serotonin target (excluding 5HT2A) at a concentration of 10 µM is minimized in addition to maximizing the predicted probability of activity against 5HT2A.

5HT2A + Dopamine & Serotonin Selectivity – The average predicted probability of bioactivity against each dopamine and serotonin target (excluding 5HT2A) at a concentration of 10 µM is minimized in addition to maximizing the predicted probability of activity against 5HT2A.

In contrast to the first set of objectives, this second set of objectives was more difficult for the generative model to optimize, as shown in Figure 2.11. The easiest objectives with respect to achieving similar scores to real 5HT2A ligands were membrane selectivity and DRD2 selectivity. The former likely due to the number of models run leading to low overall ratios of predicted off-targets. However, as more models are added, as in the dopamine and serotonin families, the objective becomes increasingly difficult to optimize to the standard of real 5HT2A ligands. With the final objective of dopamine and serotonin selectivity barely being improved throughout optimization. It is worth noting the caveat that real 5HT2A ligands are likely contained in the training data of the PIDGINv5 models used in these objectives, and so will receive inflated scores compared to 'active' unseen molecules (of which most *de novo* molecules are unseen). Although the accuracy of these models isn't known prospectively, or the maximum score achievable, the scores on real 5HT2A ligands at least provide a minimal benchmark. Moreover, the models are able to at least distinguish 95 of 126 5HT2A molecules selective over DRD2, despite the fact that 124 molecules have a DRD2 pChEMBL value of 5 or above and therefore, 93 correctly predicted selective molecules are actually false negative predictions with respect to the DRD2 model at 10 µM threshold (see Figure 2.12), which is somewhat advantageous behaviour in this case for distinguishing selective compounds. For comparison to real 5HT2A ligands selective over DRD2, I extracted the *de novo* nearest neighbours to the five most selective 5HT2A ligands (see Figure 2.13). Analogues were found in the 0.3-0.6 Tanimoto similarity range, although the identified analogues tend to be a 'simpler' version i.e., smaller with fewer heteroatoms and functional groups, indicating that either the objective or the generative model needs to more appropriately account for medicinal chemistry principles. However, the *de novo* compounds did possess similar predicted off-target profiles to the real 5HT2A ligands. Overall, this second set of objectives is a more challenging optimization problem.

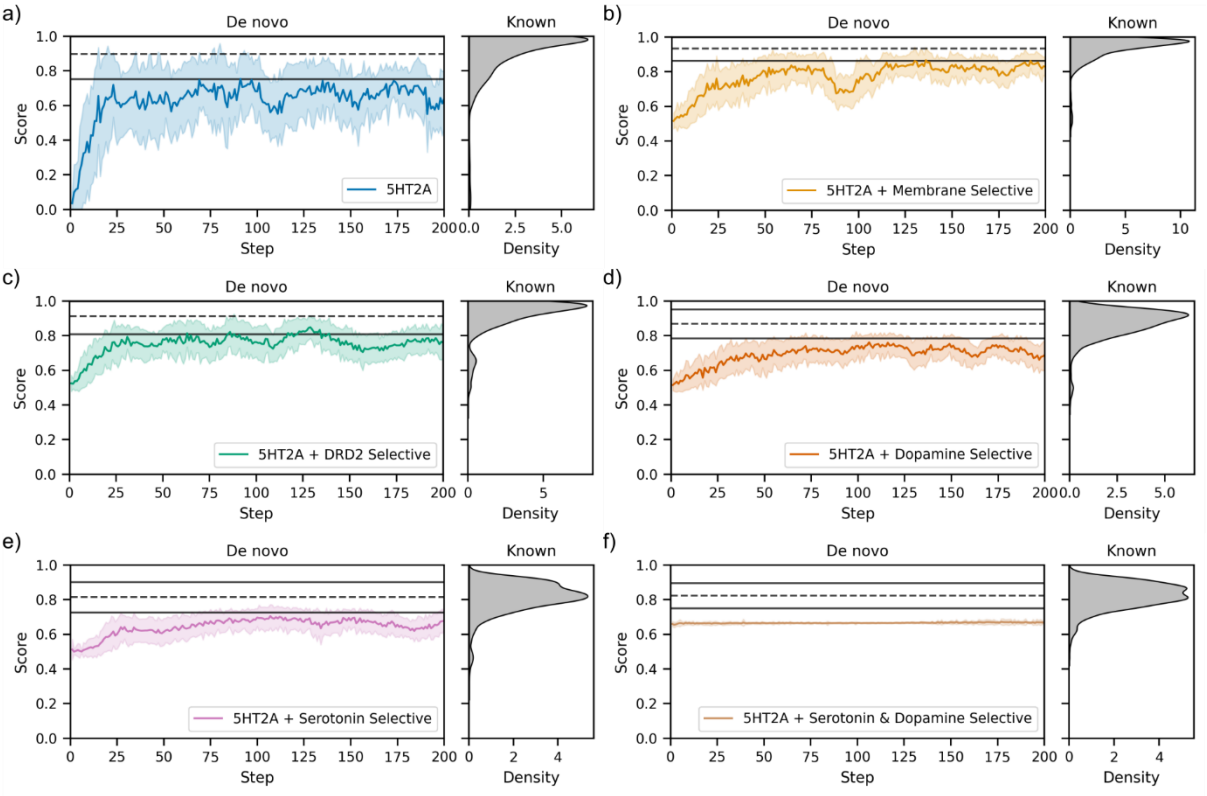


Figure 2.11: *De novo* optimization of the second set of objectives designed by molscore by number of optimization steps (left) with the equivalent score distribution for 3,771 real 5HT2A ligands (right). The dashed line represents the mean of the real ligand distribution and solid lines plus/minus one standard deviation from the mean. (a) The predicted probability of 5HT2A activity at a concentration of 1 µM. (b) The first objective (a) combined with predicted membrane selectivity. (c) The first objective (a) combined with predicted DRD2 selectivity. (d) The first objective (a) combined with predicted dopamine selectivity. (e) The first objective (a) combined with predicted serotonin sub-type selectivity. (f) The first objective (a) combined with serotonin sub-type selectivity and dopamine selectivity.

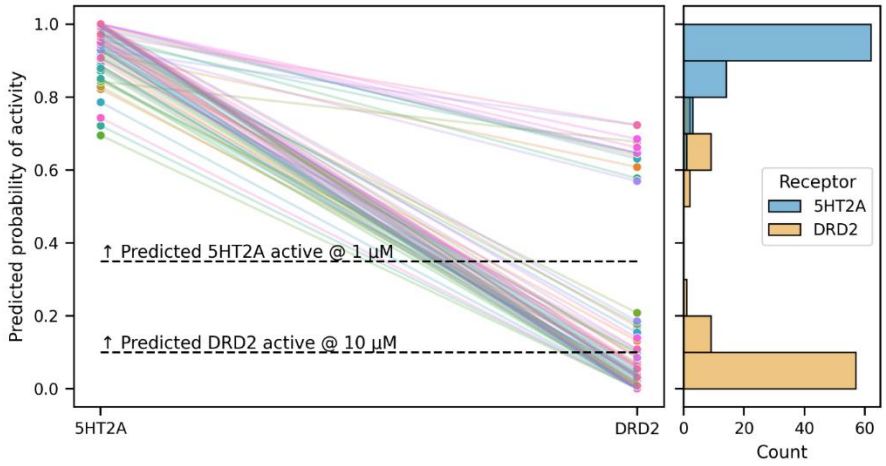


Figure 2.12: PIDGINv5 model performance on 5HT2A ligands selective over DRD2. Predicted probability of 5HT2A selective ligands (left) by 5HT2A classification model at 1 µM (all predicted active) and by DRD2 classification model at 10 µM (95 predicted inactive) and (right) their respective distribution of predicted probability. Selective ligands are defined as 5HT2A ligands extracted from ChEMBL31 with an average pChEMBL value for 5HT2A assays at least 6 and at least 2 greater than for DRD2 assays (i.e., 100-fold or greater). Dashed lines represent the optimal classification decision threshold for each respective model.

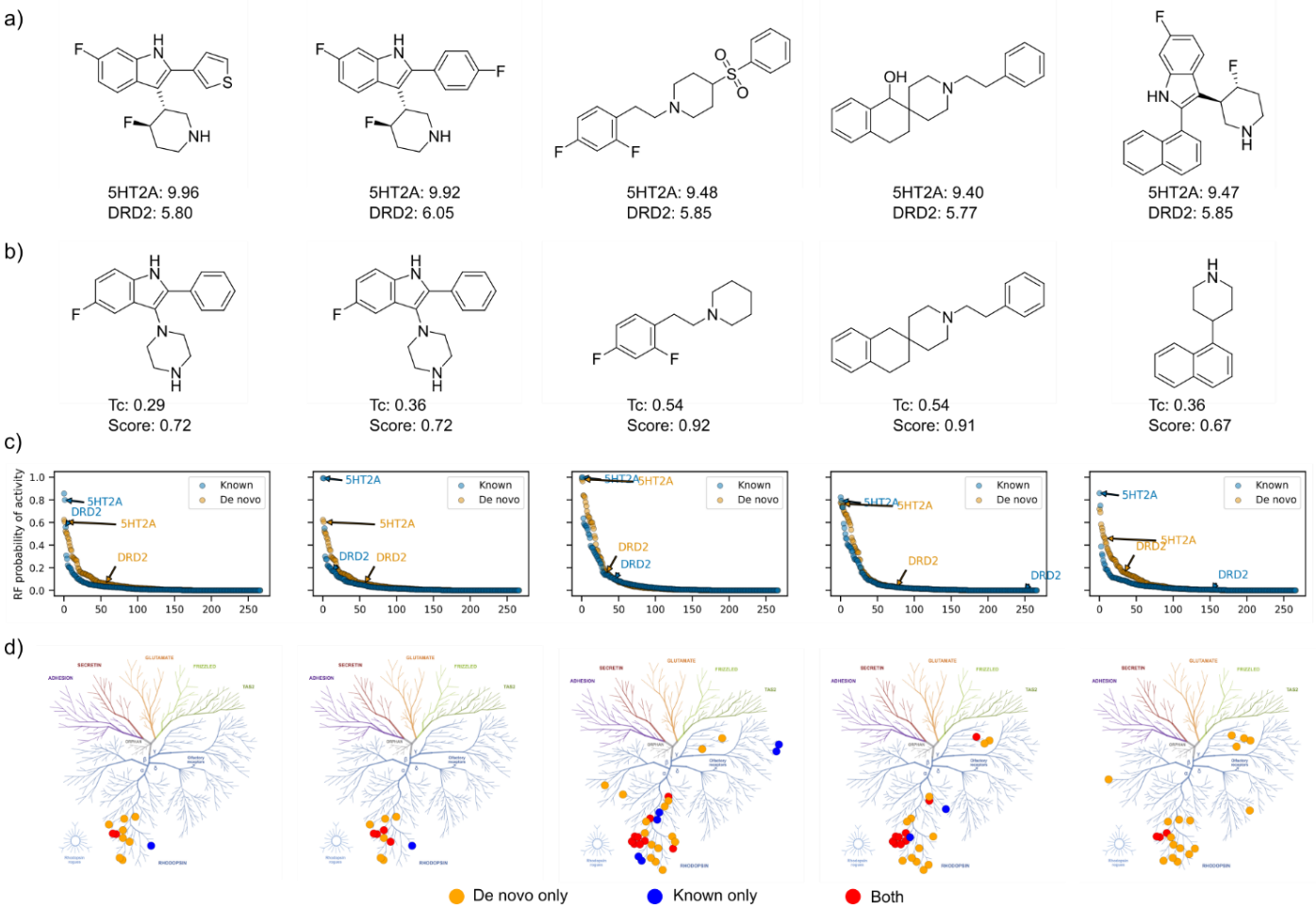


Figure 2.13: Example nearest neighbour *de novo* molecules to real 5HT2A ligands selective over DRD2. (a) The five most 5HT2A selective ligands with respect to DRD2 binding identified in ChEMBL31 that contain a DRD2 pChEMBL value above 4, respective pChEMBL values are shown. (b) Nearest neighbour *de novo* molecules to each molecule in (a), identified during the 5HT2A + DRD2 Selective task with respective Tanimoto similarity (Tc) and objective score. (c) Predicted probabilities of class A GPCR off-targets for real and *de novo* ligand counterparts using PIDGINv5. (d) Predicted class A GPCR targets mapped onto a GPCRome tree[248] , shared predicted targets are shown in red, predicted only for the real ligand in blue, and predicted only for the *de novo* ligand in orange.

2.3.4 Moleval use case: Evaluating fine-tuning epochs

The suite of performance metrics does not necessarily need to be run on a molscore output. Instead, it can be used to assess arbitrary datasets for quick comparison to reference datasets. For example, evaluating progress during generative model fine-tuning. In this case study, I use a SMILES-based RNN pre-trained on ChEMBL compounds and fine-tune it (*via* transfer learning) using a set of known A2A ligands to bias *de novo* molecule generation towards A2A-bioactive-like chemotypes. This just requires two lines of Python to instantiate the `GetMetrics` class specifying any reference datasets and calling `calculate` to calculate the metrics (in this case, repeated for sampled *de novo* molecules after each epoch of fine-tuning).

Figure 2.14 shows the resulting changes in metric values during fine-tuning where Epoch-0 represents the generative model before fine-tuning began. It is quickly possible to assess that some intrinsic properties (Figure 2.14a) like novelty and diversity decrease with increasing fine-tuning epochs, while validity has an initial drop that is recovered with further fine-tuning epochs as it adjusts to new chemotypes. Meanwhile, similarity to the initial pre-training dataset (ChEMBL compounds) decreases as shown by an increase in Fréchet ChemNet Distance[165] and decrease in analogue coverage (Figure 2.14b). Note that metrics that measure the presence of only a single similar molecule, like analogue similarity and single nearest neighbour increase, as the initial ChEMBL training dataset will likely already contain A2A-like chemotypes. Conversely, similarity to the fine-tuning set of A2A ligands increases especially noticeable by analogue similarity and coverage (Figure 2.14c), while novelty also slowly decreases with respect to this fine-tuning set. This overview of property changes allows for interpretation on how many fine-tuning epochs are required. In this case, arguably, just one or two epochs are needed which quickly leads to an increased similarity to the fine-tuning set with marginal improvements with any further epochs; however, further epochs do lead to an undesirable decrease in novelty and diversity. The required balance will vary depending on user and use case, however, quickly assessing changes is always useful.

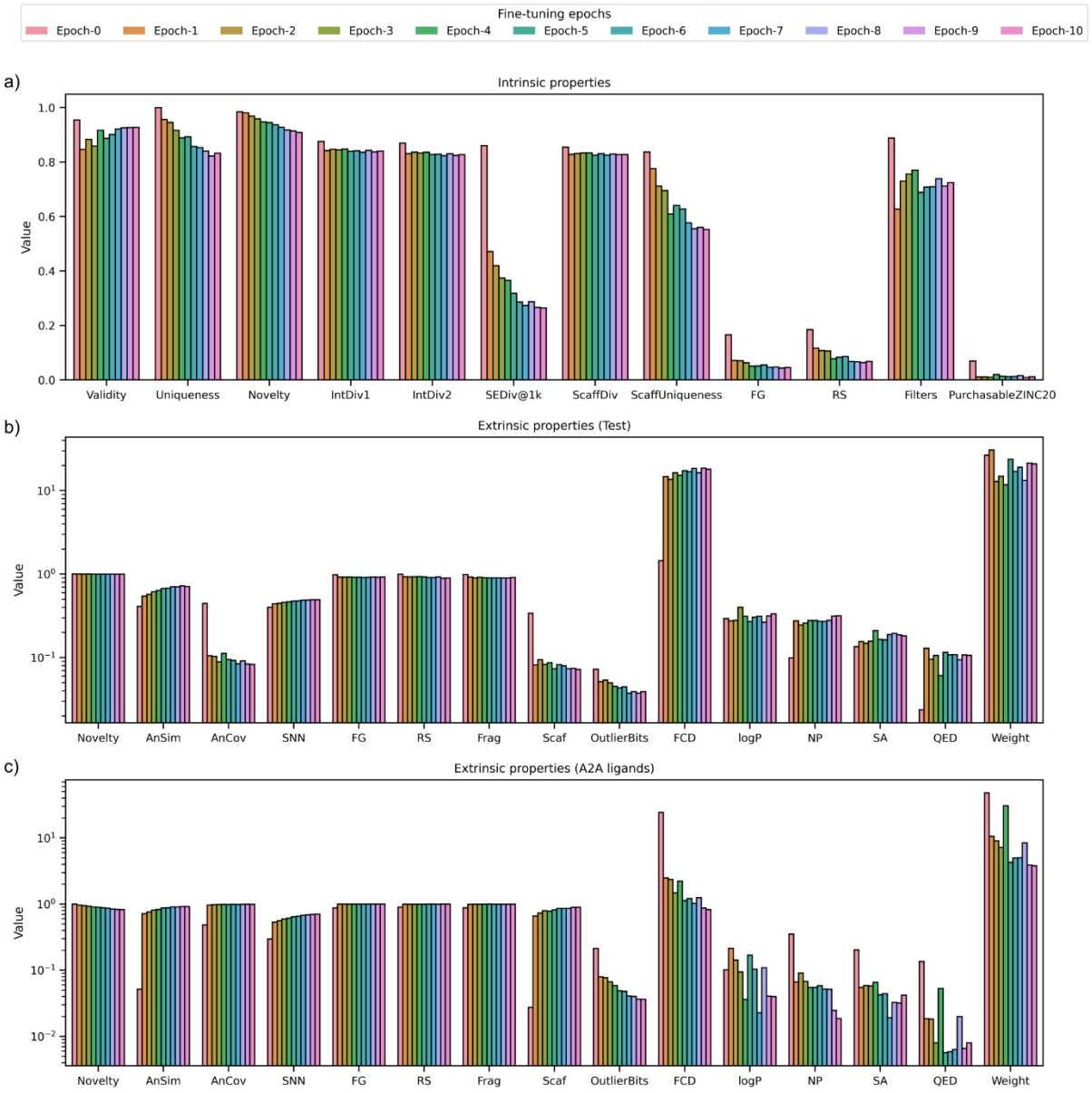


Figure 2.14: `Moleval` metrics computed on different fine-tuning epochs. Epoch-0 represents the generative model before fine-tuning. Intrinsic properties (a) and extrinsic properties in reference to a test set (sample of the training set) (b) and the set of A2A ligands used for fine-tuning (c) are shown.

2.3.5 Future developments

Aside from optimal integration of methods with conflicting library dependencies, as discussed earlier, several other improvements for MolScore are planned for the future. More generally, adding further scoring functions and performance evaluation functionality, for example, structure interaction fingerprint rescoring for docked poses. Accepting molecules with 3D conformations as inputs, particularly for structure-based scoring functions such as docking and shape alignment. This will become more useful following the increase in 3D structure-based generative models[249] and will mitigate the

need for ligand preparation protocols and conformational searches currently required for said scoring functions. Integrating dynamic configuration files that can be updated during the course of optimization for use in curriculum learning[250] (ideally the files will be modifiable directly through the monitor GUI to help inform the change in objective). Lastly, although I have not specified benchmark tasks *per se*, integrating a benchmark mode may be desirable for a particular set of objective tasks. However, I leave this for future work and hope this flexible framework can be used to design challenging benchmark objectives shared by the community.

2.4 Conclusion

MolScore introduced in this chapter is an open-source Python framework for the flexible design of difficult, drug design relevant objectives for *de novo* molecule scoring and evaluation. This framework takes a more flexible approach to generative model benchmarking, acknowledging that benchmarks will never be relevant to all situations. Instead, users can make use of the functionality available, contribute custom scoring functions and share their proposed benchmark objectives in a standardized way. In addition, this framework contains two GUIs to facilitate ease of use and accessibility. I believe this framework combines the best elements of current benchmarks with additional flexibility, leading to an overall improved platform. I introduce new performance metrics such as SEDiv and justify its use instead of standard practice, as well as demonstrate the use of MolScore to design drug design relevant objectives and how it can be used to also evaluate *de novo* molecules (and therefore differences between generative model hyperparameters, architectures and objective functions).

Chapter 3: Comparison of structure- and ligand-based scoring functions for deep generative models: a GPCR case study

3.1 Introduction

Many recent generative models have utilized ligand-based scoring functions to guide molecule design, however, ligand-based scoring functions have inherent limitations (as also discussed in 1.2.3.3). Firstly, machine learning models are restricted by their applicability domain i.e. they perform well on 'in-distribution' data but struggle to extrapolate to 'out-of-distribution' data, which is often poorly accounted for in model validation[251,252]. Ultimately, resulting in observations as seen by Renz et al.[134] where the ligand-based scoring functions biased generative models so much that *de novo* molecules were no longer predicted as active by control QSAR models, which were either initialized with a different seed or trained on a different data split. This is very likely a contributing factor to the lack of diversity and novelty seen in deep generative models[18,75,167] and a serious drawback from both an intellectual property and discovery perspective. Moreover, the use of ligand-based scoring functions also requires large enough amounts of annotated ligand data to sufficiently train a machine learning model in the first instance, which typically restricts their use in real drug discovery scenarios such as first-in-class projects, as discussed in 1.2.3.4. On the other hand, the advantages of using structure-based principles as mentioned in 1.2.2.6 are lost with ligand-based scoring functions.

This chapter is based on the hypothesis that structure-based scoring functions, as exemplified by molecular docking, may mitigate some of the limitations observed with ligand-based scoring functions. Molecular docking is a physics-based approach that uses the crystal structure (or in the absence of that a homology model) of a protein to estimate both the pose and free energy binding of a ligand[212,214,253,254]. Although the resulting free energy score is notoriously inaccurate[141,142] and the performance of these scoring functions can be highly target-dependent[178], molecular docking consistently results in the early enrichment of known active molecules in virtual libraries compared to random[141] and is a generally-applied computational ligand design method in pharmaceutical research today.

The principal advantage of the physics-based nature of molecular docking is that it is not restricted to the chemical space of existing bioactive training data from the ligand side. Provided a scoring function achieves enrichment of bioactive compounds against a protein target (which can be established on existing datasets where data is available, but where otherwise estimates can be made based on the character of the binding pocket and protein type[255,256]), then the chemical space to be scored can be greatly expanded, beyond chemistry and chemotypes present in any ligand-based training dataset. As structural input, either experimentally resolved structures or homology models can be employed and given the increasing numbers of structures available (which increases by about 10,000 per year[137]) and development of protein structure prediction technology[257], this renders this approach applicable to an increasingly wide range of protein targets.

The development of MolScore in Chapter 2 facilitates easy implementation of both ligand-based and structure-based scoring functions to be combined with a generative model. To investigate this difference between LBDD and SBDD, REINVENT[76] was chosen as a demonstrative generative model, which is a SMILES-based RNN which has evidenced competitive performance with respect to the coverage of *de novo* chemical space[104] and benchmark performances[55,80]. To understand the differences between ligand-based and structure-based scoring functions, REINVENT is either trained to maximize the probability of activity by a support vector machine (SVM) trained on known ligands, or to minimize the docking score calculated by Glide[212] using a reference co-crystal structure. The resulting *de novo* molecules are then compared.

Dopamine Receptor D2 (DRD2) was chosen as a case study. This receptor has a wealth of associated ligand bioactivity data available, and it has been commonly used in deep generative model publications before[62,76,113,167,258], thereby allowing any further comparison to different methods. DRD2 has a publicly available X-ray crystal structure[259] in complex with Risperidone, thereby allowing use of molecular docking without the requirement of generating a homology model.

At the time of conducting this work, few previous studies existed which incorporated structural data into deep generative model scoring functions, compared to the ligand-based counterpart. Firstly, Ghanakota et al.[260] combined high throughput free energy perturbation (FEP) with REINVENT to identify potential CDK2 inhibitors. To achieve

this, they trained an AutoQSAR model [261] on a subset of 1,000 enumerated analogues of a potent inhibitor with the corresponding FEP predictions, which was subsequently used as the REINVENT scoring function. The authors observed 1.5-fold enrichment selecting compounds with activity below 10 nM, compared to selecting enumerated analogues using the AutoQSAR model alone. Secondly, Li et al.[161] trained an RNN on known kinase CDK4 inhibitors and fine-tuned the network by training on a selection of generated molecules screened using docking. This was validated experimentally, with one out of nine tested molecules found to be active against the target (57.8% inhibition at 10 μM). Thirdly, Xu et al.[115] similarly used molecular docking to guide ligand selection in the latent space of a variational autoencoder towards CDK2 predicted activity, resulting in the recovery of a known CDK2 inhibitor and several molecules containing substructures of known CDK2 inhibitors. Cieplinksi et al.[80] evidenced that VAEs[30,119] were unable to generate molecules with optimized Smina[216] docking scores due to the inaccurate surrogate model prediction of said docking score, which is used to guide *de novo* sampling in the respective methods. Although, the authors propose a docking benchmark on which REINVENT outperforms the above methods and baselines of both random and known active molecules[81]. Lastly, Boitreaud et al.[126] recently used a novel sampling approach combined with a graph to SELFIES[40] variational autoencoder, where the authors demonstrated the ability to optimize the Vina[254] docking score against Dopamine Receptor D3, while maintaining chemical diversity.

Notable contrasts in this approach compared to the above approaches include: (1) Only structural data is required, enabling the search of a much larger chemical space compared to the use of ligand data as in[161]. (2) An RNN combined with RL is used as opposed to a VAE as in[126]. (3) Docking scores were used directly during the generative model optimization process, as opposed to predicting the outcome *via* a surrogate model as in[80,260]. (4) In this approach, the model parameters are updated such that new chemical distributions can be learned as opposed to sampling a fixed latent space as in[115].

3.2 Materials and methods

Figure 3.1 depicts the approach taken for the comparison of a structure- and ligand-based scoring functions in a deep generative model setting undertaken in this work.

First known DRD2 actives (according to the ExCAPE-DB[262]) were removed from the MOSES curated[79] ZINC[145] database of small drug-like molecules for use as training data. The REINVENT framework[76] was used as a deep generative model. This framework consists of two RNNs – a Prior and an Agent (see 1.2.2.5). The Prior is trained to learn the conditional probability distribution of SMILES tokens in the training dataset and the Agent is then initialized as an exact copy of the Prior. The scoring functions used in this work utilized structural data from the PBD[137] and the docking program Glide, or ligand data extracted from ExCAPE-DB[262] to build an SVM-based bioactivity model[76] to score molecules that have been generated *de novo*. The Agent then samples *de novo* SMILES strings which are subsequently evaluated by the scoring function, and the Agent is updated *via* reinforcement learning to optimize either the docking score or the predicted probability of activity. Finally, *de novo* molecules are evaluated with respect to several different quantitative, chemical and structural aspects.

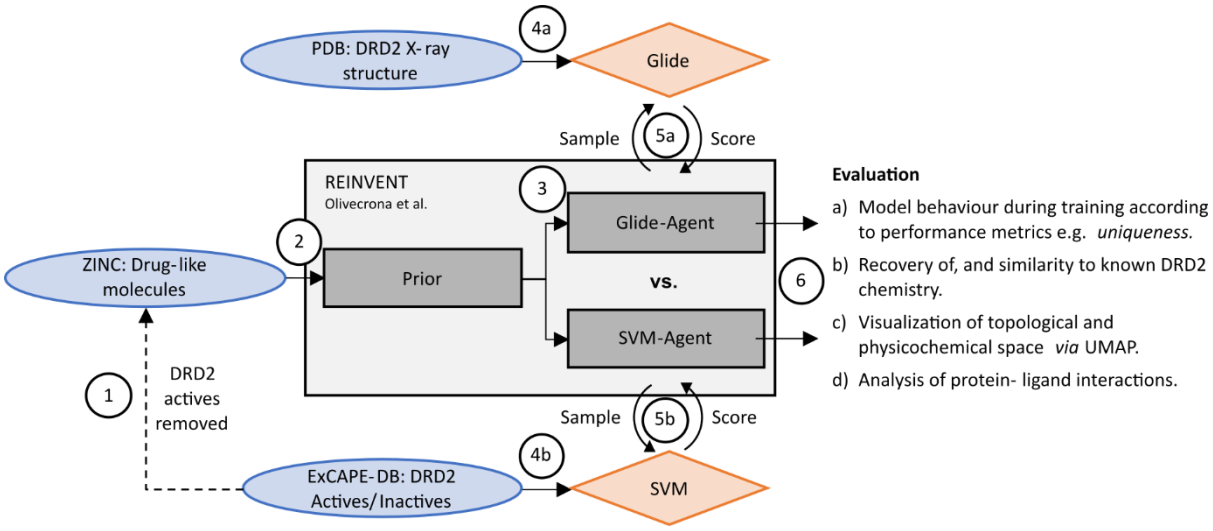


Figure 3.1: Schematic of the ligand-based *versus* structure-based comparison in this chapter including data sources (blue), scoring functions (orange), and the deep generative model framework REINVENT[76] (grey) . Main steps are (1) removing known DRD2 active molecules from the ZINC training data; (2) Training the Prior model on drug-like molecules from ZINC; (3) Initializing the Agents as a copy of the Prior; (4) Preparing the scoring functions to evaluate *de novo* molecules; (5) Iteratively training both Agents *via* reinforcement learning; and (6) evaluating the structure- and ligand-based approach with respect to different quantitative, chemical and structural aspects of the generated molecules.

3.2.1 Datasets

The dataset used to train the Prior network was modified from the curation described by MOSES[79], in which the authors extracted molecules as SMILES from the ZINC15 database[145]. In short, molecules were selected to adhere to the following rules:

molecular weight between 250-350 Da; number of rotatable bonds not greater than 8; XlogP[263] not greater than 3.5; no charged atoms; no atoms besides C, N, S, O, F, Cl, Br, H; no cycles larger than 8 members; custom medicinal chemistry filters[264,265]; and finally PAINS filters[7] were applied. I deviate from this curation by first allowing charged atoms and then neutralizing protonatable groups. This was achieved by modifying the MOSES pipeline, implemented using RDKit[199], to remove the filter that checks for formal charge on atoms and instead add/remove protons to neutralize atoms where possible[266]. As a result, the training split contains 2,454,087 molecules as opposed to 1,584,664 described in the publication[79]. The authors rationalized the charge filter as such, “we removed charged molecules to avoid ambiguity with tautomer’s and pH conditions. Note that in the initial set of molecules, functional groups were present in both ionized and unionized forms.”. However, given the nature of AR molecule generation conditional upon the rest of the molecule using RNNs used in this work, this filter could remove relevant chemical structures in which the ‘functional group - whole structure’ conditional relationship may not be duplicated. In addition, only ~6,500 charged variants are also present in the neutral form in the ZINC15 subset out of the ~870,000 removed due to the charge filter. This may further lead to a bias towards non-protonatable chemical structures which are crucial for aminergic receptors as used in this work, as aminergic receptors typically require an ionic interaction with a conserved aspartic acid residue in the orthosteric site (Ballesteros-Weinstein: $D^{3.32}$, GPCRdb: D^{3x32})[267,268]. To further require the RNN to explore novel chemical space, any canonical SMILES that matched the canonical SMILES of any known DRD2 active molecules extracted from the ExCAPE-DB[262] were also removed (as canonicalized by RDKit[199]). This resulted in a training set of 2,454,048 canonical SMILES, I call MOSES^{n}.

In order to generate a set of bioactive compounds with known DRD2 activity, molecules from ExCAPE-DB[262] were extracted. ExCAPE-DB is a curation of ChEMBL20[204] and PubChem[269] data that classifies molecules with a measured dose-response value equal to or lower than 10 μM as active, and with higher than 10 μM (or those which were labelled inactive in the original sources) as inactive. This resulted in 4,613 active and 343,028 inactive molecules against human DRD2. However, as it may be unreasonable to expect the generative model to generate molecules outside the property space on which it was trained, the same filtering as previously described

is applied to create another subset labelled 'in'. In addition, for use as a reference baseline a random set of molecules with the same filters applied were extracted from ChEMBL26[204]. Resulting in the following subsets with size: Active (all), 4,613; Active (in), 396; Inactive (all), 10,000; Inactive (in), 10,000; Random 10,000.

The DRD2 X-ray crystal structure 6CM4 bound to Risperidone from the PDB[137] was used as the protein structure for docking.

3.2.2 REINVENT

The training data described in *Datasets* was subject to further filtering in accordance with the REINVENT pipeline[76] to standardize SMILES input, tokenize SMILES symbols and construct a vocabulary for one-hot encoding. This filtering resulted in 2,453,916 unique, non-isomeric (stereochemistry removed) SMILES that was subsequently used to train the Prior network for a total of 5 epochs with a batch size of 128 using the Adam optimizer [270] with a learning rate of 0.001. The Agent was then trained for 3,000 steps using a batch size of 64 and the Adam optimizer with a learning rate of 0.0005 and a value for the scalar coefficient (σ) of 60. These hyperparameters were used as recommended by the publication[76] and not explored further. All neural network training was conducted on an NVIDIA RTX2080$_{Ti}$ GPU.

3.2.3 Scoring functions

A ligand-based scoring function was used as a baseline. An SVM model previously published by Olivecrona et al.[76] trained on 7,218 active and 100,000 inactive DRD2 molecules, which were also extracted from ExCAPE-DB[262], is used. Note that this figure differs from the human DRD2 bioactives used for evaluation in this chapter. It is likely that the authors did not filter bioactive molecules by species (as it stands this would result in 7,919 active DRD2 molecules without further processing[262]), which however is particularly important in the current work due to the use of the human ortholog of DRD2 for docking, and hence particular attention is paid to this here. The resulting SVM predicts the uncalibrated probability of a molecule to be active against DRD2.

The structure-based scoring function used protein-ligand docking. The DRD2 crystal structure was prepared using the Schrodinger Protein Preparation Wizard[271] using default parameters i.e. hydrogens added, non-residue molecules protonated (e.g.

ligand, cofactors) at pH 7±2 using Epik[219], hydrogen bond assignment optimized at pH 7 using PROPKA[272] and the structure minimized using the OPLS3e force field[273]. Any waters, cofactors, or crystallisation artefacts (e.g., oleic acid) were removed from the structure. A grid was defined using the centroid of the co-crystallised ligand Risperidone as the centre. From the ligand side, before docking, molecules were prepared using LigPrep[218], enumerating unspecified stereocentres, tautomers and protonation states (using Epik[219]). Up to 8 variants were prepared per molecule based on a pH range of 7±1 and minimised using the OPLS3e force field. Each molecule and any respective variants were then docked using Glide standard precision (GlideScore-SP[212]) with default settings, flexible ligand sampling, standard precision with Epik state penalties, post-docking minimization of five poses and final output of the single best scoring pose. Using between 36 and 50 CPUs, the wall time required for 3,000 iterations was approximately 7 days, based on an average scoring time of 3 mins per 64 molecules (including molecule preparation and up to 512 individual docking runs including respective variants).

3.2.4 Retrospective validation of docking protocol and scoring functions

In the REINVENT study, the authors evaluated the performance of the SVM model on an undisclosed held-out test set, resulting in an accuracy of 98%, precision of 97% and recall of 82%.

To also evaluate the performance of the docking protocol, all 4,613 known DRD2 active molecules and a random subset of 10,000 DRD2 inactive molecules were docked. The performance of classification into either active or inactive molecules at various docking score thresholds was then investigated (see Figure 3.2) according to classification accuracy, precision, and recall (which can be calculated as shown in Equation 3.1 based on the number of true positives TP, true negatives TN, false positives FP and false negatives FN. A docking score of -7.5 resulted in highest overall accuracy of about 76%. By decreasing the threshold to -8.5 (i.e., a more stringent criterion for selecting active molecules), a higher precision of approximately 82% is achieved, although at lower accuracy of about 74% and lower recall of about 12%. However, the latter more stringent threshold might still be a more favourable one to use in practice, given that confidence in positive predictions of active compounds is often more relevant than missing some active compounds (of which there are many)

due to low recall. It should be remembered that the performance of the scoring function was not an objective in its own right (given that retrospective evaluations naturally favour ligand-based methods due to analogue bias in databases etc.[274]), but rather to ensure general suitability for the desired purpose of selecting active compounds in this step.

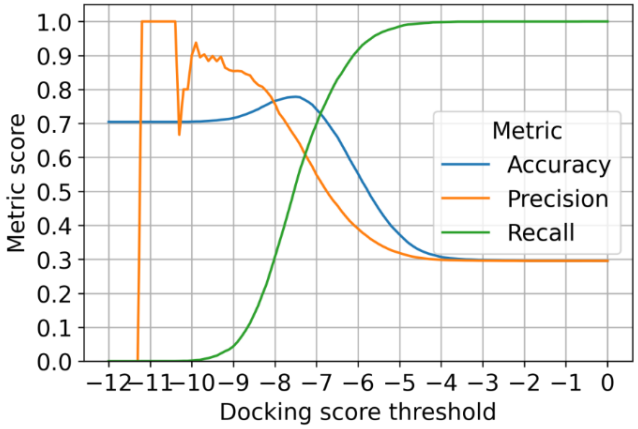


Figure 3.2: Retrospective performance of the docking protocol as a classification problem. Retrospective performance of the Glide docking protocol on known human DRD2 active and inactive molecules extracted from ExCAPE-DB. The docking score is used as a decision threshold to predict molecules as active or inactive, and the accuracy, precision and recall are reported at a variety of docking score decision thresholds. It can be seen that a docking score threshold of -8.5 results in a precision of approximately 82%.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$
$$Precision = \frac{TP}{TP + FP}$$
$$Recall = \frac{TP}{TP + FN}$$

Equation 3.1

3.2.5 Clustering

Molecular clustering was performed on molecules or their respective Bemis-Murcko scaffolds[170] using the sphere exclusion algorithm[163] as implemented by Roger Sayle[275] in RDKit[199]. The sphere radius was set at a Tanimoto distance of 0.65 and 0.2 for molecules or their respective scaffolds using Morgan fingerprints (*radius=2, nBits=1024*). These parameters were chosen based on the same used in the similarity analysis conducted by Sayle[275]. Once resulting sphere centroids had been picked, molecules were assigned to the nearest centroid to form a cluster.

3.2.6 Chemical space visualization

In order to further understand the chemistry generated by both approaches (and their distribution across chemical space), Uniform Manifold Approximation and Projection (UMAP)[276] was performed using both molecular fingerprint and property descriptors for representation, as well as calculating the normalized principal moments ratio (NPR)[277]. For the former, Morgan fingerprints (*radius*=2, *nBits*=1024, implemented using RDKit) of actives (either 'in' or 'all'), Prior, Glide-Agent and SVM-Agent molecules were used as input features, and the UMAP was calculated using the Jaccard distance metric with a minimum distance 0. For property space, the CLogP, molecular weight, heavy atom count, # H-bond acceptors, # H-bond donors, # Heteroatoms, # rotatable bonds, # aromatic rings, # aliphatic rings, # rings, topological polar surface area, fraction of sp^3 carbons, QED[173] and SAscore[172] were calculated using RDKit and scaled before input to UMAP using default parameters. Lastly, the NPR1 and NPR2 were calculated using RDKit after first generating 3D conformations using the ETKDG method[278].

3.2.7 Structure interaction fingerprints

Structure Interaction Fingerprints (SIFts)[279] were calculated on all resulting docked poses in order to understand ligand-protein interactions available to the generated ligands. This resulted in a 9-element bit vector for each protein residue, corresponding to non-exclusive residue interactions. For simplification, the non-exclusive 9-element bit vector (comprising the possible interactions any contact, backbone, sidechain, polar, hydrophobic, hydrogen bond acceptor, hydrogen bond donor, aromatic, charged) is converted to exclusive residue interactions in a hierarchical manner according to the following order: charged hydrogen bond donor/acceptor, hydrogen bond donor/acceptor, charged, aromatic, hydrophobic/polar. For example, a residue initially defined as having sidechain, polar, charged and hydrogen bond acceptor interactions would be converted to charged hydrogen bond acceptor, due to this interaction type taking precedent in the above order. This simplification was performed to allow for more interpretable (and less redundant) subsequent analysis of the interactions observed.

3.3 Results and discussion

3.3.1 Optimization of SVM- and Glide-Agent-based scores by molecules generated de novo

First, I investigated whether the Agents were able to optimize the respective properties evaluated by the two scoring functions i.e., predicted probability of DRD2 activity based on bioactivity data ('SVM-Agent') and DRD2 docking score ('Glide-Agent'), the results of which are shown in Figure 3.3. Both the SVM-Agent and Glide-Agent learn to generate molecules with optimized properties, albeit at different rates. Whilst the SVM-Agent converges to generating optimal molecules within just a few hundred steps, the Glide-Agent only begins to converge after about 2,000 training steps. Crucially, both Agents maintain high ratios of valid (> 0.9) and novel molecules per batch (> 0.9). However, from just 100 steps onwards, the SVM-Agent starts to generate fewer unique molecules than the Glide-Agent (Figure 3.3d). This suggests mode collapse, as the SVM-Agent has maximally optimized the scoring function and begins to re-sample molecules that it knows produce a high reward. This is further supported by a drop in the diversity of sampled molecules and their scaffolds (Figure 3.3f-h). SEDiv indicates that after 200 steps the chemical space of SVM-Agent *de novo* molecules can be explained by less than 10% of the valid and unique molecules, while for the Glide-Agent this slowly drops to about 20%. In addition, the SVM-Agent shows an increased FCD[165] to a held out test set with respect to the Glide-Agent (Figure 3.3i). This increase in FCD has shown to indicate a number of differences[165] to the training data for example, 'drug-likeness' defined by QED[173] or IntDiv[162]. In addition to performance metrics, and similar to Blaschke et al.[167], the cumulative number of analogues generated *de novo* to known DRD2 active molecules was investigated (see Figure 3.4). This analysis shows that the SVM-Agent generates more analogues (~80,000) than the Glide-Agent (~25,000), however, when instead looking at the number of DRD2 active molecules with generated analogues, the Glide-Agent has analogues to more DRD2 actives (~1,800) than the SVM-Agent (~ 1,400). Thus, the SVM-Agent generates more analogues per known active, but the Glide-Agent generates analogues to a broader range of known actives. Together, these results indicate that the Glide-Agent maintains better performance throughout training, in particular with respect to the uniqueness and general diversity of the generated molecules. Also, the Glide-Agent generates analogues to more known DRD2 active

molecules, further evidencing increased diversity also with respect to known DRD2 active molecules.

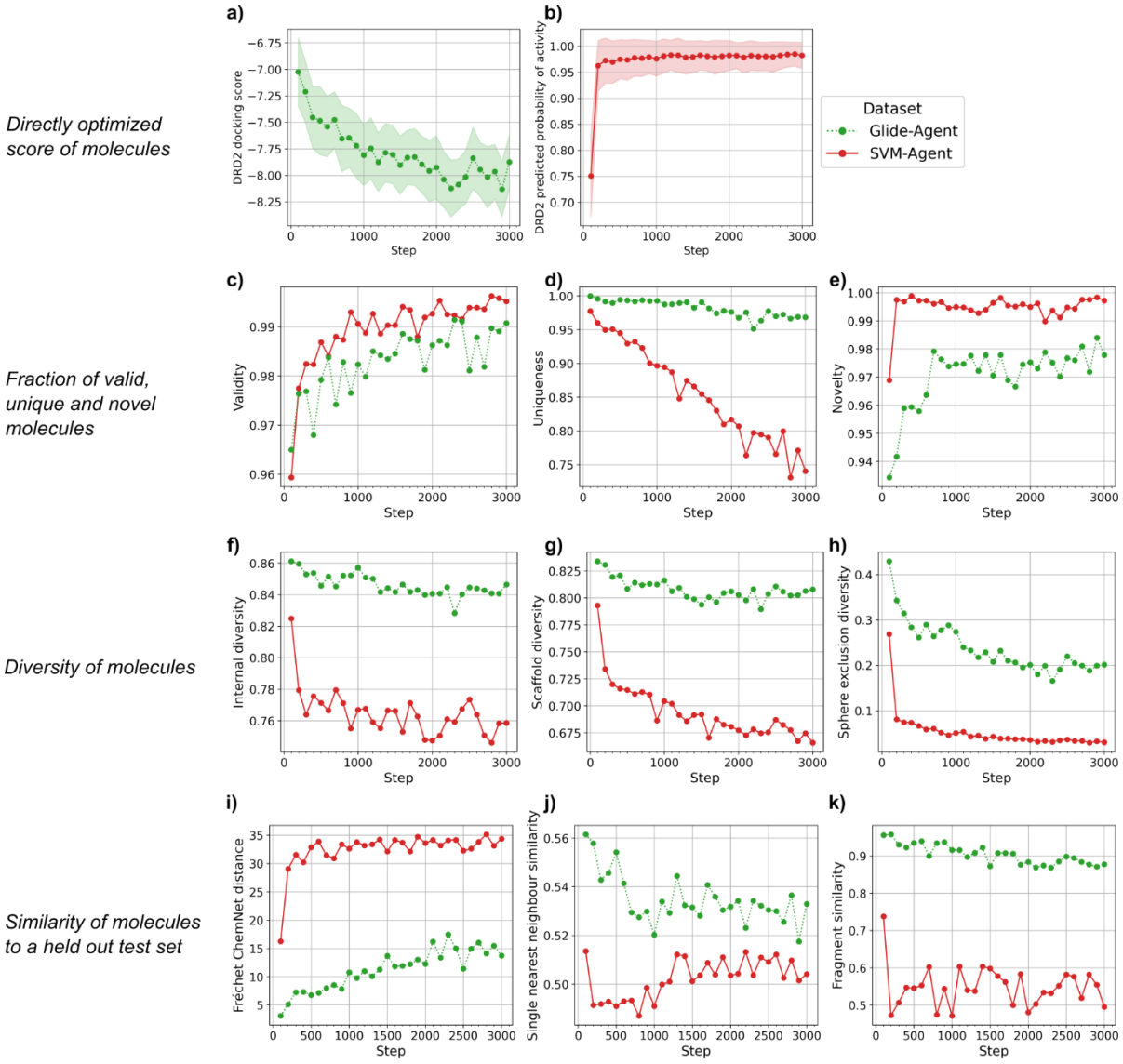


Figure 3.3: Generative model performance during optimization for the Glide-Agent (green) and the SVM-Agent (red), calculated every 100 steps. Mean optimization of scores - docking score and predicted probability of activity - are shown in (a) and (b) respectively, as well as the 95% confidence interval. Additional metrics shown are (c) validity, (d) uniqueness, (e) novelty, (f) internal diversity, (g) scaffold diversity, (h) sphere exclusion diversity, (i) Fréchet ChemNet Distance, (j) single nearest neighbour similarity and (k) fragment similarity. As the most important observation, the SVM-Agent reaches very high scores much more quickly, which comes at the cost of a significant reduction in uniqueness and diversity, when compared to the Glide-Agent. For definitions and detailed discussion see main text.

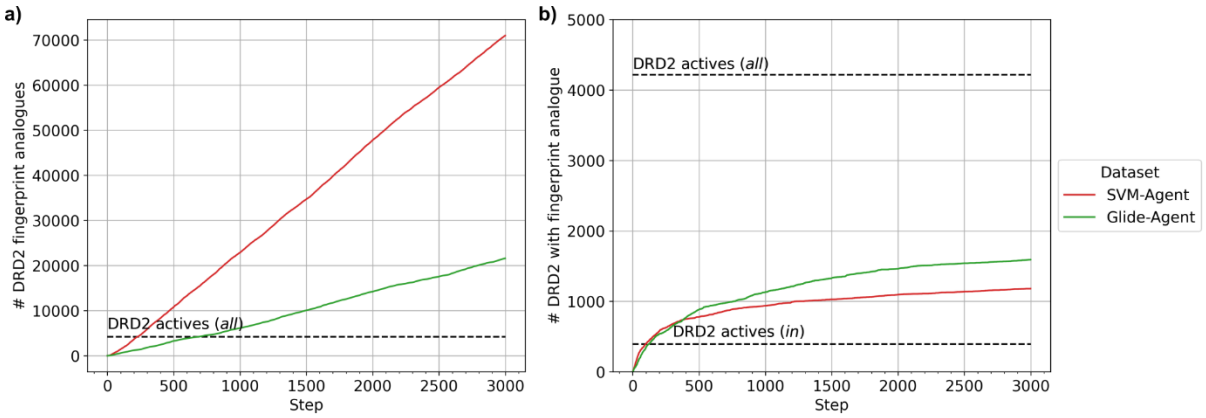


Figure 3.4: *De* novo DRD2 analogues generated during optimization. (a) The cumulative number of molecular fingerprint analogues to known DRD2 active compounds and (b) the number of known DRD2 active molecules with analogues generated during optimization. The SVM-Agent generates more analogues to known DRD2 active molecules, although, the Glide-Agent generates analogues to more known DRD2 active molecules.

For any generative model, visual inspection of the generated molecules is crucial, both to see whether an approach tends to prefer different types of chemistry, and to identify any possibly idiosyncratic behaviour. In this regard, Figure 3.5 displays the centroid of the largest clusters generated during training, as well as the respective cluster size. This shows that the chemotypes evolve from the Prior differently depending on the scoring function. Overall, both Agents were able to optimize molecules towards their respective scoring functions (as shown quantitatively in Figure 3.3); however, the Glide-Agent does so with more diversity (Figure 3.3f-h) and with a more similar distribution to the training data (Figure 3.3i-k).

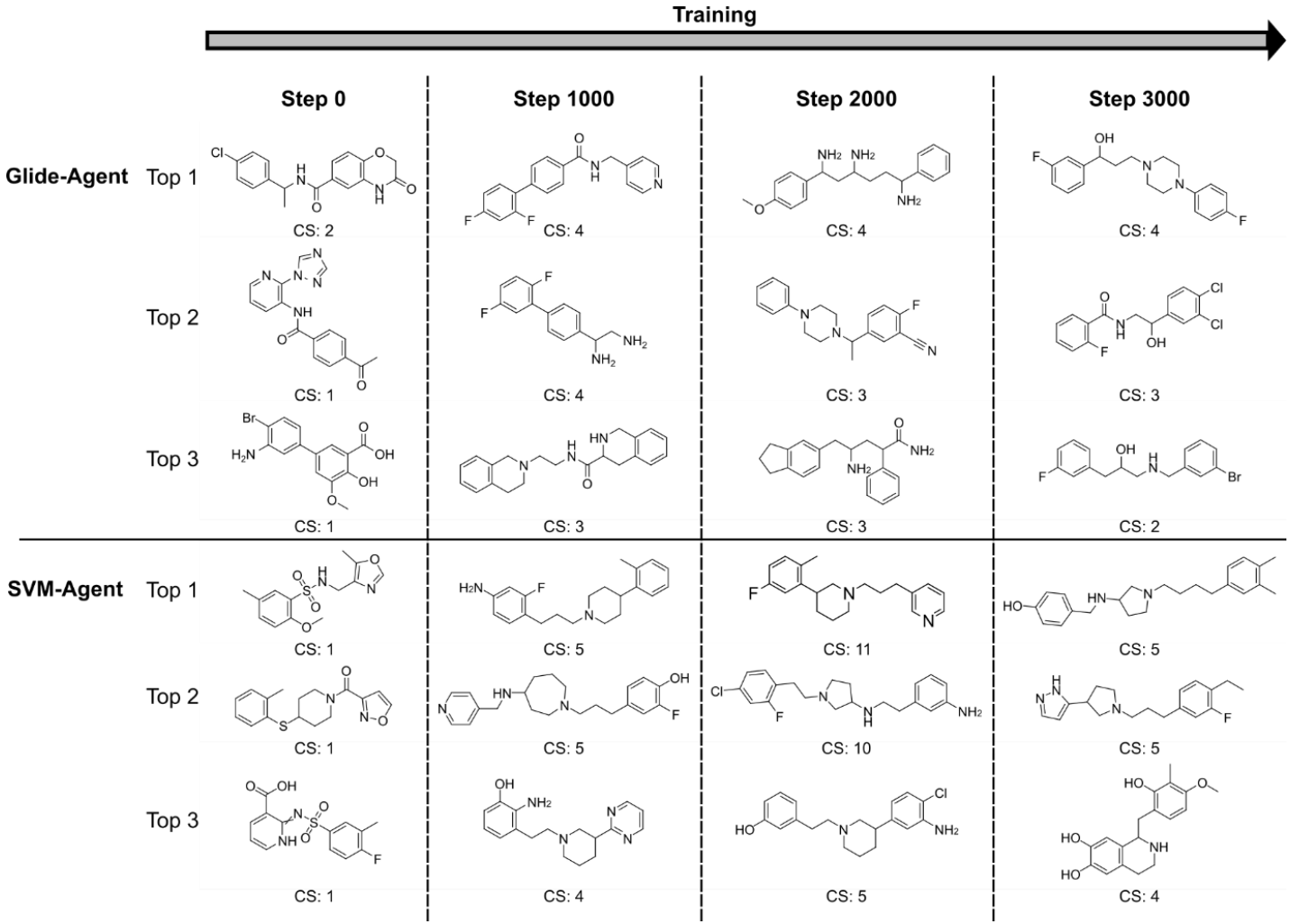


Figure 3.5: Chemotype evolution during training, comparing the SVM-Agent and the Glide-Agent. Molecules were sampled during training at the start and after 0, 1,000, 2,000 and 3,000 steps (at the end of training). Molecules in each batch were clustered and the centroids of the three largest clusters are shown here, alongside respective cluster size (CS). This visualizes the difference in topology and chemotype between the two approaches.

For further analysis, 10,000 molecules were sampled from the unoptimized Prior, the SVM-Agent (trained for 500 steps, before significant overfitting occurred), and the Glide-Agent (trained for 2,000 steps). MOSES performance metrics[79] were calculated on the generated molecules as well as, scaffold diversity, scaffold uniqueness and SEDiv (see 2.2.2.1). Coinciding with the results observed in Figure 3, Table 3.1 to Table 3.3 show that the Glide-Agent outperforms the SVM-Agent in all metrics except novelty. Overall showing greater diversity of *de novo* molecules and similarity to the training data (whilst still optimizing the docking score).

Table 3.1: Basic generative model metrics of the Prior, Glide-Agent (@2000 steps) and SVM-Agent (@500 steps).

Model	#	**Valid** (↑)	**Unique** (↑)	**# valid & unique** (↑)	**Novelty** (↑)	**Filters** (↑)
Random	10000	1.0	1.0	10000	0.720	0.938
Train	10000	1.0	1.0	10000	0.0	0.999
Prior	10000	0.988	1.0	9879	0.800	0.995
SVM-Agent	9979	0.990	0.897	8865	**0.995**	0.964
Glide-Agent	**9993**	**0.990**	**0.953**	**9434**	0.978	**0.967**

Table 3.2: Diversity metrics of the Prior, Glide-Agent (@2000 steps) and SVM-Agent (@500 steps).

Model	**IntDiv$_1$** (↑)	**IntDiv$_2$** (↑)	**SEDiv** (↑)	**SEDiv @1k** (↑)	**ScaffDiv** (↑)	**Scaff uniqueness** (↑)
Random	0.874	0.868	0.440	0.809	0.857	0.757
Train	0.863	0.856	0.366	0.753	0.844	0.687
Prior	0.863	0.857	0.386	0.756	0.844	0.699
SVM-Agent	0.752	0.741	0.044	0.124	0.720	0.293
Glide-Agent	**0.831**	**0.821**	**0.123**	**0.337**	**0.797**	**0.381**

Table 3.3: Similarity metrics of the Prior, Glide-Agent (@2000 steps) and SVM-Agent (@500 steps) to training and held out test data.

Model	**FCD** (↓)			**SNN** (↑)		**Frag** (↑)		**Scaff** (↑)	
	Train	Test	TestSF	Test	TestSF	Test	TestSF	Test	TestSF
Random	3.110	3.109	3.269	0.544	0.517	0.980	0.977	0.474	0.208
Train	0.124	0.129	0.660	0.645	0.584	1.000	0.998	0.865	0.000
Prior	0.133	0.138	0.643	0.614	0.565	1.000	0.998	0.850	0.076
SVM-Agent	35.975	35.952	35.977	0.504	0.492	0.502	0.499	0.040	0.005
Glide-Agent	**16.462**	**16.445**	**17.533**	**0.528**	**0.502**	**0.840**	**0.828**	**0.252**	**0.075**

Next, to better understand the extent to which the docking score could be optimized relative to known DRD2 active molecules, all *de novo* molecules were docked and their docking scores compared to the active, inactive and random reference dataset. The actives and inactives are further split into 'all' molecules extracted from ExCAPE-DB and molecules 'in' a similar chemical space as imposed by the same filters applied to the training data. To compare distributions, an adjusted one-tail t-test was conducted to only detect significant improvement in docking score distributions. Figure 3.6a shows the docking score distribution of the Glide-Agent *de novo* molecules (μ = -8.05, σ = 0.95) is significantly enriched (adjusted $p < 0.05$) over unoptimized Prior molecules (μ = -6.17, σ = 1.02) and importantly also over previously known DRD2 active molecules (μ = -7.45, σ = 1.01)(adjusted $p < 0.05$), especially those after filtering to impose similar chemical space restrictions (μ = -6.96, σ = 0.74)(adjusted $p < 0.05$). In other words, the Glide-Agent *de novo* molecules are predicted to be often as active, and on average even more active, than known DRD2 active molecules according to the Glide docking protocol. If the precision for selecting active molecules for retrospective docking at a score threshold of -8.5 translates also prospectively to *de novo* generated molecules, 32.70% percent of the Glide-Agent *de novo* molecules are predicted to be active against DRD2 (that is with a dose-response value lower than 10 µM), compared to 19.98% percent of SVM-Agent *de novo* molecules and 0.54% percent of Prior *de novo* molecules (which is relatively close to experimental hit rates that would be expected by chance alone, for example Xiao et al.[280] had an experimental hit rate against DRD2 of ~0.6%). Interestingly, the SVM-Agent *de novo* molecules also exhibit a significant enrichment (adjusted $p < 0.05$) in docking score distribution (μ = -7.85, σ = 0.80) beyond known DRD2 active molecules, although to a lesser extent. This docking score distribution enrichment is hypothesized to be a factor of generating similar *de novo* chemistry to known DRD2 actives and hence, a docking score enrichment is observed. However, the improvement over known actives may also be due to an element of randomness, as Renz et al. observed different chemical space occupation for independent runs with similar models[134]. Furthermore, a previous run conducted resulted in a smaller enrichment for the SVM-Agent but an almost identical enrichment for the Glide-Agent (data not shown). The predicted probability of DRD2 activity according the SVM for all reference datasets (Figure 3.6b) shows that most known DRD2 actives and the SVM-Agent *de novo* molecules are predicted active with high probability (0.9-1.0). Unlike docking, which predicts SVM-Agent molecules

to be equally as, or more active than known DRD2 active molecules, the SVM does not predict many Glide-Agent molecules to be active (about 75% with a low predicted probability of 0-0.1). Due to the limitations of such machine learning models, this could be evidence of a limited applicability domain. This is supported by the greater SNN similarity of the SVM-Agent *de novo* molecules to DRD2 actives that were used to train the SVM model by Olivecrona et al.[76] (see Figure 3.7). Overall, the docking score of *de novo* molecules can generally be optimized by this Glide-based agent, and this is true even beyond the scores of known active molecules.

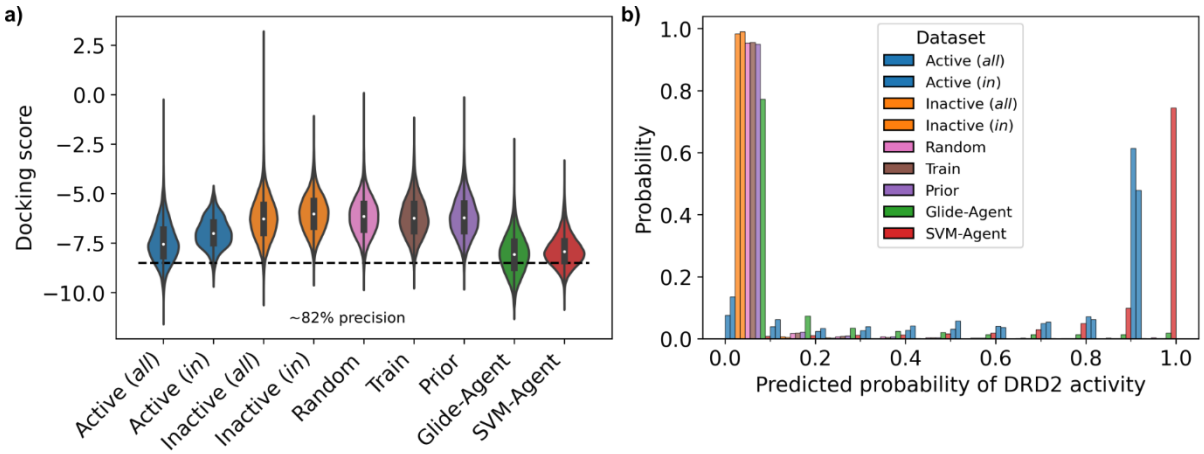


Figure 3.6: Score optimization compared to reference datasets. (a) Docking scores and (b) predicted probability of DRD2 activity of molecules generated *de novo* using the Prior, the SVM-Agent and the Glide-Agent, compared to the active, inactive, and random reference datasets. The more negative the docking score, the better it is predicted to bind. The Glide-Agent generated molecules have the best docking score distribution, more so than known DRD2 active molecules, whilst the SVM-Agent generated molecule distribution is more similar to known DRD2 active molecules. The SVM-Agent molecules and known DRD2 actives score most highly according to the SVM, comparatively, the Glide-Agent molecules do not.

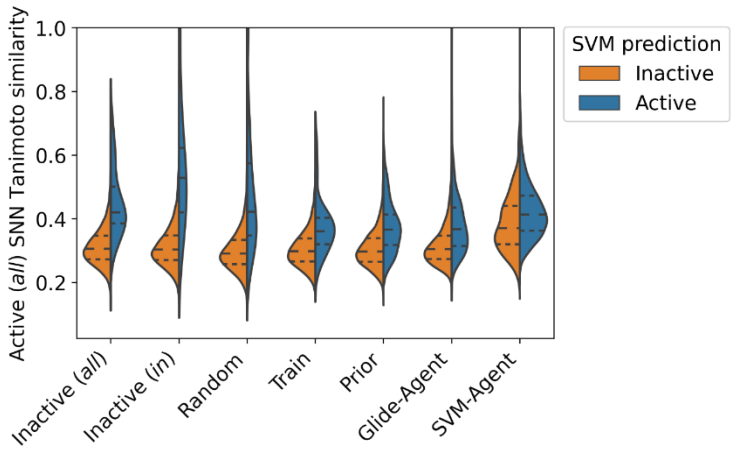


Figure 3.7: Single nearest neighbour (SNN) similarity to DRD2 active molecules by SVM predicted activity. The SVM-Agent molecules have a greater mean SNN similarity to DRD2 active molecules

than the Glide-Agent molecules and predicted active molecules are more similar than predicted inactive.

3.3.2 Overlap analysis of molecules generated de novo compared to known active and inactive molecules

To assess recovery of known active molecules, I identified whether any of the canonical SMILES produced by either Agent matches those of known DRD2 active molecules. The number of recovered molecules across ten samples of 10,000 molecules was converted into the probability of recovery (based on valid and unique molecules generated). It is worth noting that the Prior has an inherent bias towards generating inactive molecules over active molecules. The bias is quantified as the probability of generating a known active molecule over the probability of generating a known inactive molecule. This translates as the Prior being 0.002 times as likely to generate an active molecule compared to an inactive (which is partly also due to removing known DRD2 active molecules from the training data). When considering recovery of 'all' extracted DRD2 actives and inactives, both Agents are still biased towards generating inactive molecules; however, the SVM-Agent improved 95-fold towards generating active molecules over the Prior. This bias shift is predominantly attributable to the SVM-Agent's ability to avoid recovering known inactive molecules (approximately half the probability than the Glide-Agent), whereas the probability of recovering known active molecules is more comparable between the Glide- and SVM-Agents ($63x10^{-6}$ vs $79x10^{-6}$, respectively). It is important to consider that Glide docking does not incorporate any prior knowledge of known DRD2 active and inactive molecules (unlike the SVM), and therefore the Glide-Agent is able to learn to recover known active molecules (and improve the Prior bias 40-fold) from the information of protein structure alone. Interestingly, of the single sample of 10,000 molecules investigated throughout this work, there are no recovered active molecules in common between the Agents, and just three in total (see Figure 3.8), further underlining their divergent behaviour. In summary, both Agents can similarly recover known DRD2 active molecules, however, the SVM-Agent is more adept at not generating known inactives and thus provide different types of molecules generated *de novo* as a result.

Table 3.4: Probability of recovering known DRD2 active and inactive molecules. The reported probability values are the mean (and standard deviation) across ten samples of 10,000 *de novo* molecules drawn from the model, and the bias towards generating active molecules over inactive molecules (and fold change from Prior) on the right. The Glide- and SVM-Agent have a similar probability of recovering known active molecules, therefore the SVM-Agent bias towards generating active molecules over inactivate molecules is mostly driven by the lower probability of generating inactive molecules.

Origin of dataset	**Probability of generating active molecule (x10^{-6})**		**Probability of generating inactive molecule (x10^{-6})**		**Active bias (fold change from Prior)**	
Active DRD2 chemical space relative to training data	*In*	*All*	*In*	*All*	*In*	*All*
Prior	10 (30)	10 (30)	5055 (604)	5957 (495)	0.002 (1)	0.002 (1)
Glide-Agent	11 (32)	63 (84)	422 (125)	917 (175)	0.025 (12.5)	0.069 (40.6)
SVM-Agent	34 (72)	79 (72)	256 (124)	486 (168)	0.130 (64.9)	0.163 (95.7)

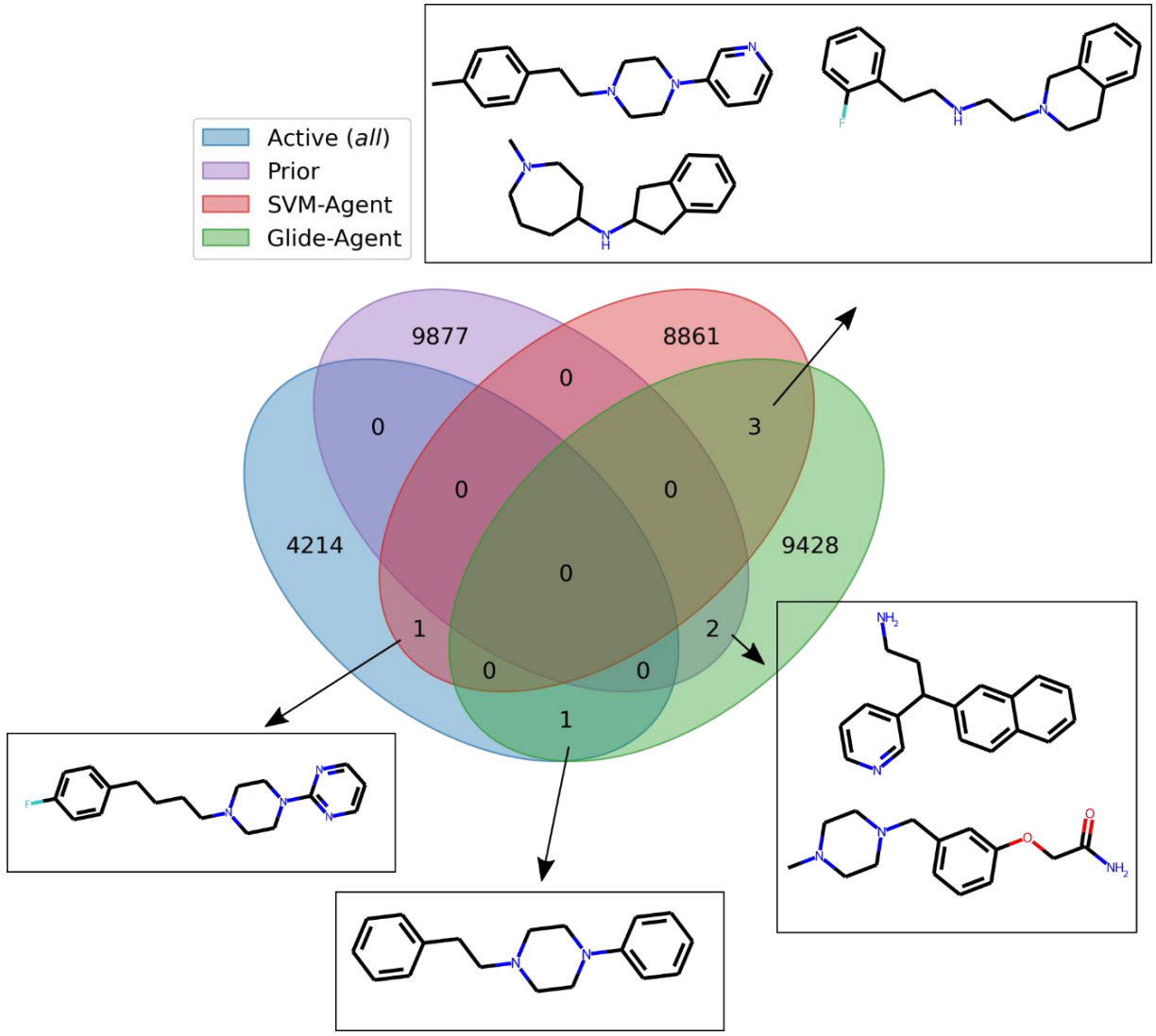


Figure 3.8: Overlap of *de novo* molecules to DRD2 active compounds. Each Agent only managed to recover one active molecules, and in total only shared three *de novo* molecules between them - exemplifying divergent behaviour.

3.3.3 Similarity analysis of molecules generated de novo to known active and inactive molecules

The DRD2 analogue analysis conducted during training was repeated, investigating the number of analogues to known DRD2 active compounds as in Blaschke et al.[167]. Similar to the results observed during training, the SVM-Agent sample contains a higher fraction of molecules considered fingerprint analogues to DRD2 actives (both to actives 'in' a similar chemical space and 'all' extracted). Furthermore, both Agent samples contain a higher fraction of analogues to DRD2 actives than inactive molecules (which one would expect to be relatively high based on the chemical series nature of drug design). However, the Glide-Agent generates analogues to a higher fraction of DRD2 actives, indicating that the higher diversity observed (Table 3.2) is relevant with respect to active chemistry. In addition, the DRD2 actives with *de novo* analogues differed depending on the Prior or Agent (see Figure 3.9), evidencing complementary behaviour with respect to identifying similar molecules to known actives.

Table 3.5: Fraction of molecules that are fingerprint analogues to DRD2 active molecules and respective fraction of DRD2 active molecules with analogues. The SVM-Agent generates more analogues to known actives, however, the Glide-Agent generates analogues to more known actives, demonstrating a greater coverage of known active space.

Origin of dataset	**Fraction of *de novo* molecules with DRD2 active analogues**		**Fraction of DRD2 actives with *de novo* analogues**	
DRD2 chemical space relative to training data	*In*	*All*	*In*	*All*
Inactive *(in)*	0.020	0.089	0.197	0.116
Inactive *(all)*	0.025	0.102	0.242	0.116
Train	0.020	0.071	0.225	0.109
Random	0.024	0.075	0.313	0.120
Prior	0.021	0.071	0.220	0.110
Glide-Agent	0.051	0.124	0.268	0.105
SVM-Agent	0.179	0.563	0.237	0.102

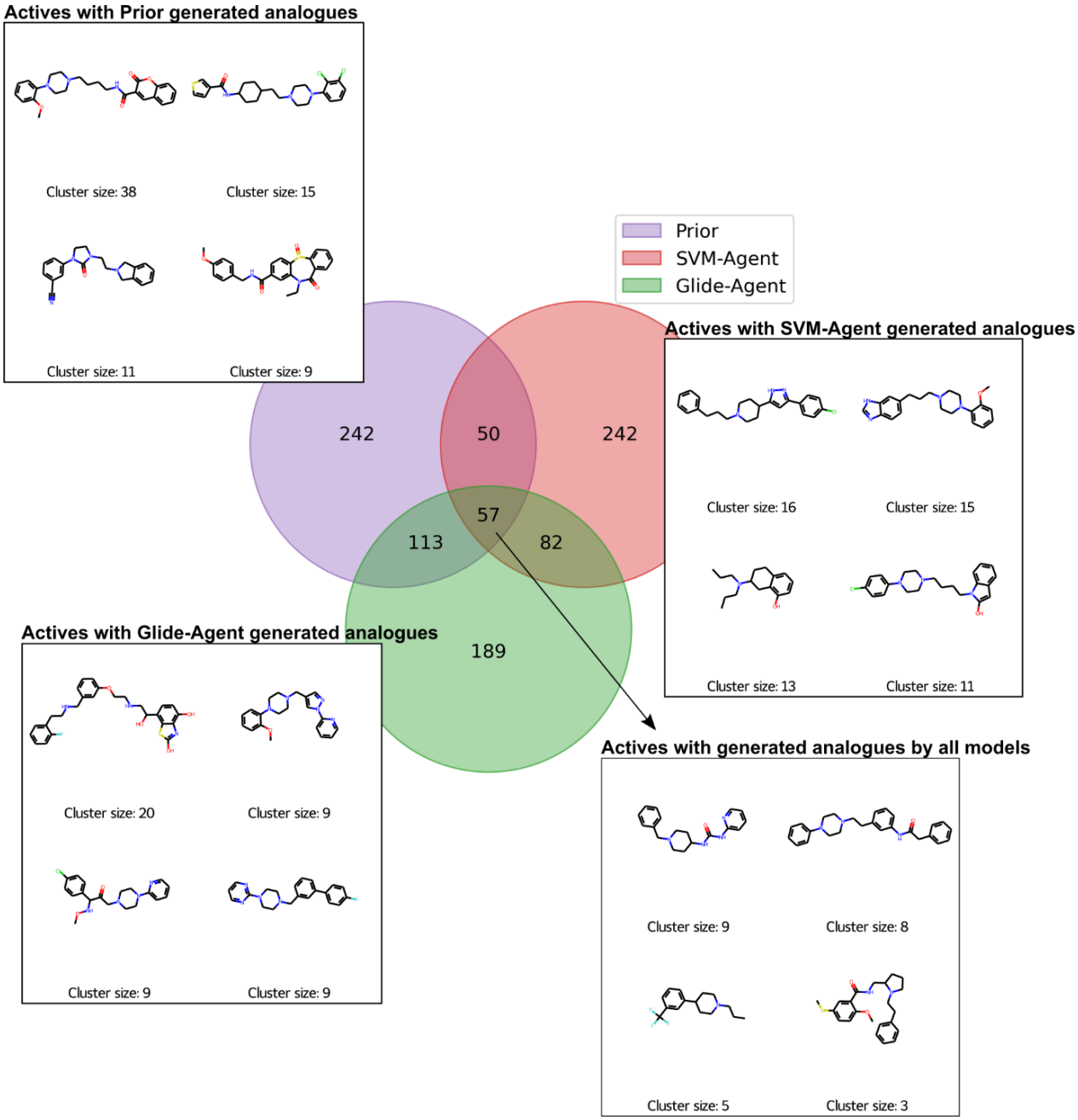


Figure 3.9: Overlap between Active (all) molecules with analogues generated by generative models. Centroids of the largest clusters are shown to represent the most common chemistry present in selected subsets. The DRD2 actives with analogues generated are different depending on the specific Agent. Suggesting these approaches are complementary to each other for discovering similar active molecules.

How similar the *de novo* generated molecules are to known DRD2 active molecules and/or each other is also of interest. Therefore, known DRD2 active molecules were clustered together with the Prior, Glide- and SVM-Agent *de novo* molecules. Each cluster was then analysed to identify to which dataset each of its members belonged (similar to Tomberg et al.[281]). Figure 3.10 shows the results of this analysis as a Venn diagram for both entire molecules (Figure 3.10a) and their respective Bemis-Murcko scaffolds (Figure 3.10b). This analysis shows more clusters – 105 – are shared between known active DRD2 molecules and the Glide-Agent, compared to the overlap of known active DRD2 ligands with the SVM-Agent, where this number is 95. This is also observed when clusters are calculated based on scaffolds (49 vs 39 respectively). To qualitatively assess cluster behaviour, examples of clusters and structures are shown in Figure B.1 and Figure B.2. Overall, both the Glide-Agent and SVM-Agent share a relatively similar number of clusters (i.e. 'chemical space pockets') with known DRD2 actives, but which precise clusters are shared differs largely between both Agents.

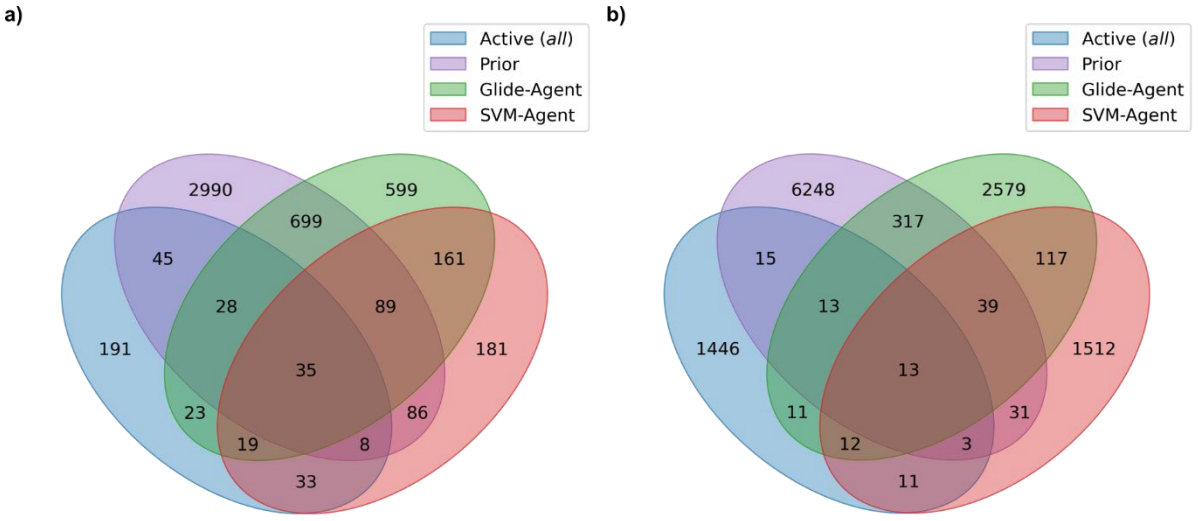


Figure 3.10: Chemical space overlap between the Prior, SVM- and Glide-Agents with all DRD2 ligands extracted from ExCAPE-DB. Broader clusters (a) were defined by clustering molecules with a Morgan fingerprint Tanimoto similarity to a centroid of 0.35 or greater, while narrower clusters (b) were defined by clustering molecules on their Bemis-Murcko scaffold Morgan fingerprint Tanimoto similarity to a centroid of 0.8 or greater (examples shown in Figure B.1 and Figure B.2). Numbers specify the number of clusters with at least one member belonging to an annotated dataset. For example, there are 23 clusters (a) where each cluster has at least one member belonging to DRD2 actives and Glide-Agent molecules. Both the Glide-Agent and SVM-Agent share clusters with known DRD2 active molecules.

3.3.4 Novelty of de novo molecules relative to known DRD2 active molecules

Similarity comparisons of *de novo* molecules to known molecules with desirable properties can provide a measure of confidence that a model is in the correct chemical space. However, prospective use case ultimately requires structural novelty to known compounds with activity against the same biological target. Figure 3.11 shows that the Glide-Agent generated molecules that have enriched docking scores below the retrospective threshold of -8.5 also have lower SNN to known DRD2 active molecules than the SVM-Agent and Prior molecules. Therefore, the Glide-Agent molecules are also more novel with respect to known actives than the SVM-Agent molecules. This could prove very important in the early stages of hit discovery.

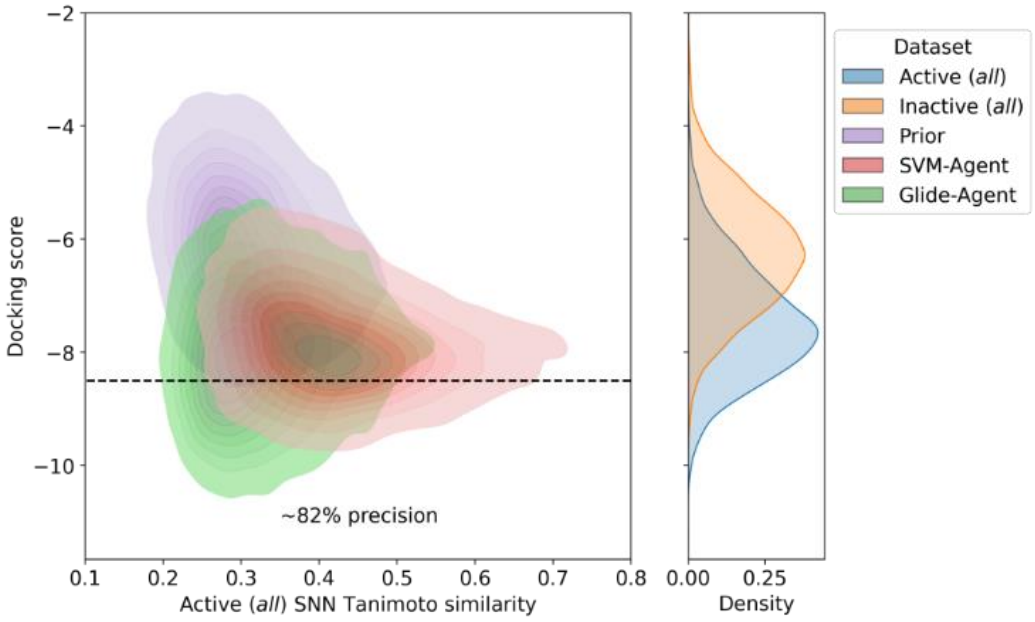


Figure 3.11: Kernel density estimates of the bivariate distribution of docking score and single nearest neighbour similarity to known DRD2 active molecules. The Glide-Agent distribution contains a shoulder with lower (better) docking scores at lower similarity to known actives than the SVM-Agent and Prior de novo molecules.

3.3.5 Differences in chemical substructural and physicochemical property space between Glide- and SVM-Agent generated molecules

To further understand the chemical differences between the molecules generated by the Glide- and SVM-Agent Uniform Manifold Approximation and Projection (UMAP) [276] was used to reduce the molecular fingerprint and physicochemical and property descriptor-based representations of chemical structures into two dimensions for visualization purposes. Furthermore, the 3D shape of molecules is investigated by the

normalized principal moments ratio (NPR)[277]. Figure 3.12 shows the two-dimensional embedded space of known DRD2 active molecules (filters applied to impose similar chemical space), as well as Prior, Glide- and SVM-Agent generated *de novo* molecules. When molecules are defined by their molecular fingerprints (Figure 3.12a), the Glide- and SVM-Agents occupy different regions of chemical space, of which neither have significant distribution overlap with known DRD2 active molecules. The SVM-Agent *de novo* molecules are more distinct from the Prior molecules, albeit still restricted by nature of the optimization function and inclusion of the Prior likelihood. In Figure 3.12b, where molecules are defined by physicochemical and property descriptors, the Prior and Glide-Agent *de novo* molecules occupy a complementary and more diverse area of property space than SVM-Agent molecules. Annotating molecular properties highlights that the clustering predominantly correlates with the number of H-bond donors and the number of aromatic/aliphatic rings (see Figure 3.13). Figure 3.12c shows a smaller difference in the distribution of 3D shapes between the datasets, again the models show slight complementary behaviour where the Glide-Agent distribution stretches slightly more towards spherical shapes and SVM-Agent slightly more towards disk shapes, although this difference is minor. The observations seen here are similar when considering ‘all’ DRD2 actives extracted from ExCAPE-DB (see Figure B.3), however, the representation is compressed due to larger and more distinct molecules seen in the active set. This analysis further corroborates, in a visual manner, the chemical differences between the structure- and ligand-based approaches, and the additional physicochemical diversity obtained by the Glide-Agent, which is not biased towards the properties of known bioactive molecules.

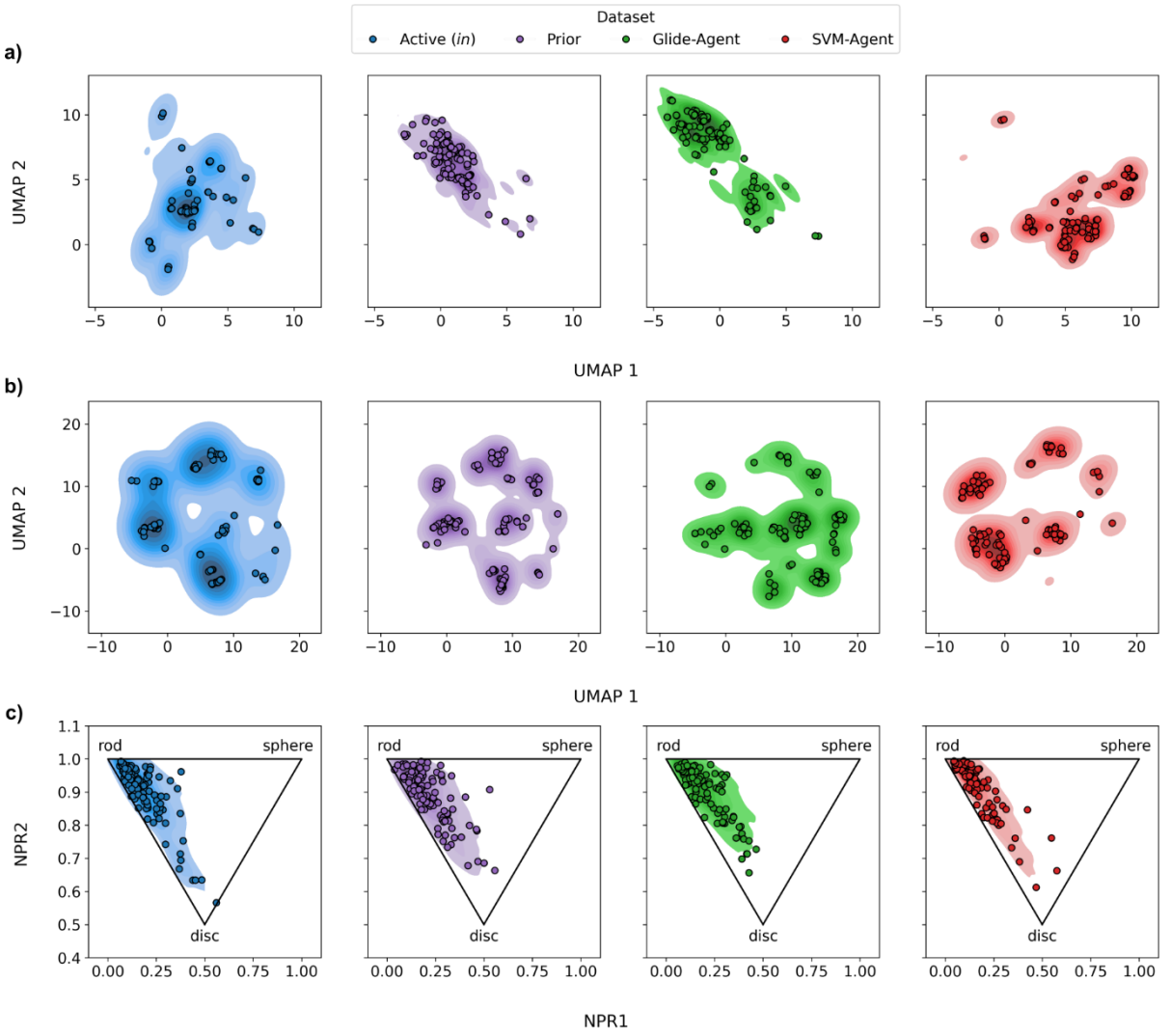


Figure 3.12: Chemical space representation of (a) molecular fingerprints and (b) physicochemical descriptors and (c) 3D space via moments of inertia. The plots show the calculated kernel density estimate with 100 randomly drawn samples overlayed. UMAP representation (a-b) was calculated for known active DRD2 ligands with filters applied to impose a similar chemical space, as well as the chemical structures associated with the Prior, Glide- and SVM-Agents. The Agents occupy complementary regions of topological space (a), physicochemical property space (b) and slightly 3D space (c) (where the Glide-Agent stretches slightly more towards spherical and the SVM-Agent slightly more towards disc shape).

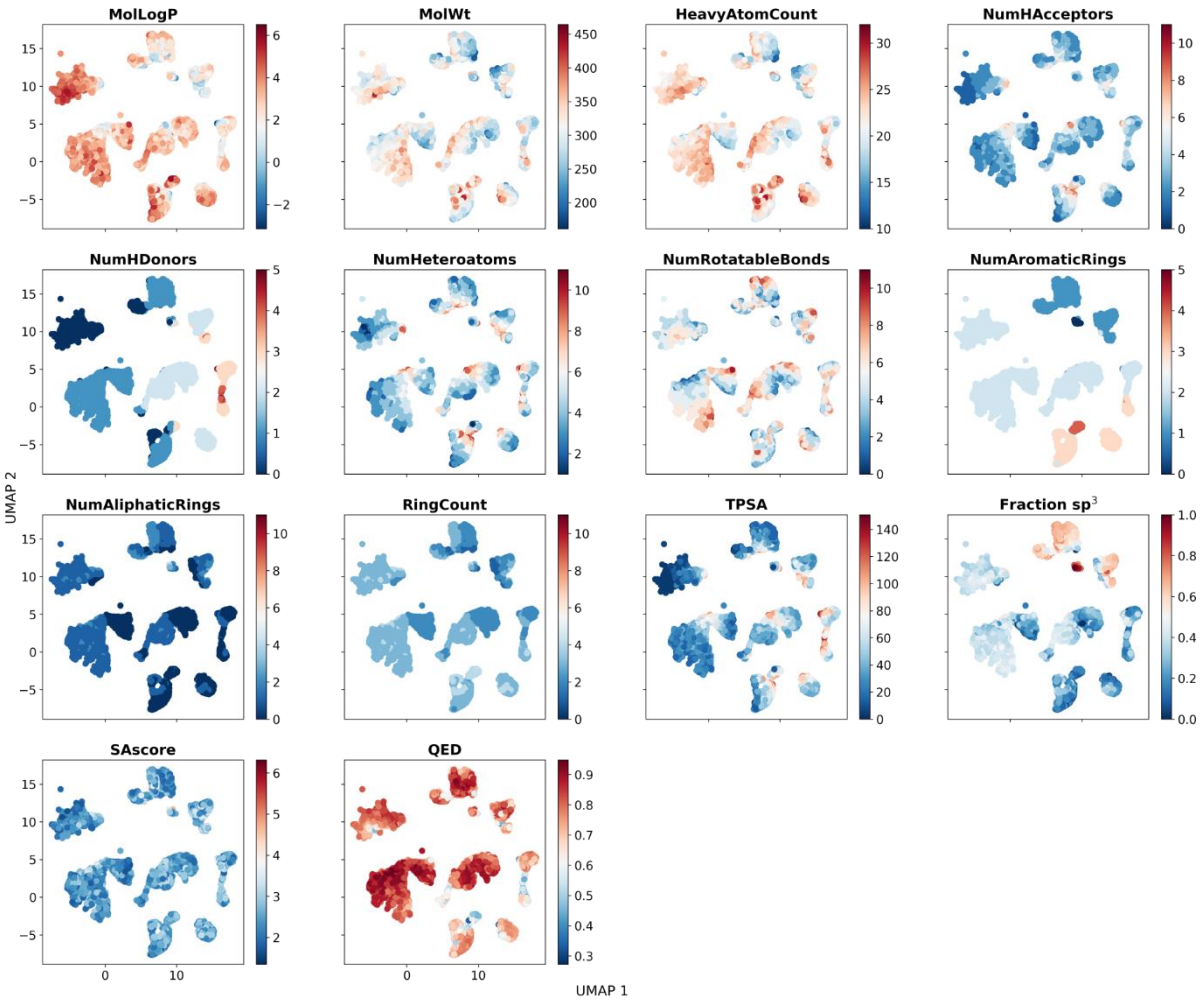


Figure 3.13: UMAP representation of physicochemical space as shown in Figure 3.12 annotated by physicochemical property descriptors used to calculate the embedding. It can be seen that the properties that most correlate with the clustering are the number of H-bond donors and ring counts.

3.3.6 Characterization of ligand chemistry obtained *de novo* chemistry

In order to understand the occupation of chemical space of optimized Agents on a ligand structural level, the molecules in each dataset were clustered according to their Bemis-Murcko scaffolds[170] which resulted in more stringent clusters more akin to chemical series. When filtering out clusters with less than 10 members (i.e., smaller 'virtual series'), the Glide-Agent set contained more clusters with better mean docking scores than all other datasets (see Figure 3.14). More specifically, the Glide-Agent set contains 30 such clusters with a mean docking score less than the previously defined threshold of -8.5, compared to just six clusters of DRD2 actives, 22 in SVM-Agent set and zero clusters in the Prior set. In this way, the Glide-agent was able to identify chemical series that dock consistently well; something that is less frequently observed for the SVM-Agent or even known actives, and non-existent for Prior *de novo*

molecules. This behaviour is analogous to the identification of bioactive chemical series in an experimental screening, where additional confidence is provided that the compounds identified are indeed true positive hits, as opposed to singletons, as false positives can occur due to experimental error (or, in the current case, idiosyncratic behaviour of the scoring function). Alternatively, it could be argued that the scoring function is not sensitive enough to identify subtle differences in ligand chemistry that result in inactivity, commonly referred to as activity cliffs i.e. strong nonadditivity in structure-activity relationships. However, one study investigated strong nonadditivity between matched molecular pair cycles with respective structural data[282], and identified that in 10 out of 15 possible cases there was a structural explaination, such as, complete rearrangement of binding mode or substiuent interactions causing nonadditivity. Therefore, in theory scoring functions that take into account structural information may better account for nonadditivity than purely ligand-based ones.

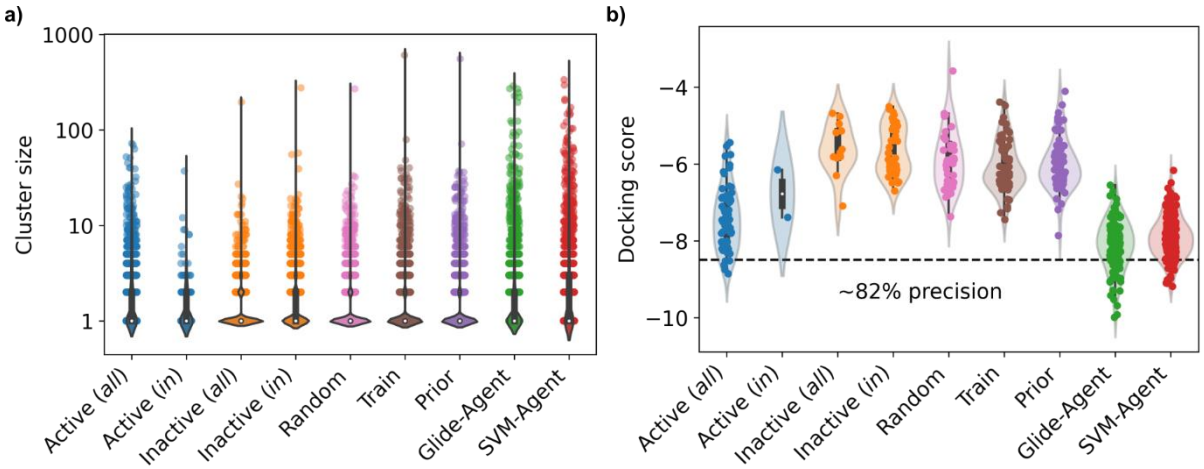


Figure 3.14: Size and docking score of molecular clusters. Distribution of molecular clusters (a) according to their cluster size and (b) docking score for those with a size greater than or equal to 10. Docking score reflects the mean docking score of all molecules in a cluster. The Glide-Agent contains more clusters of size 10 or more, with lower (better) mean docking scores.

Figure 3.15 shows the cluster centroids of the two largest and the two best-scoring clusters from each respective dataset (minimum of 10 clusters). Typical known DRD2 active molecules are ‘capped’ by mono- or bicyclic systems which are linked by an aliphatic chain that usually (but not exclusively) contains a piperidine/piperazine moiety. This chemotype is not well recapitulated by the Prior molecules as it is not optimized towards DRD2 bioactivity in any way. The Glide-Agent on the other hand learns to mostly cap the molecules with mono- or bicyclic systems, but it does not generate the piperidine/piperazine moiety in the compounds shown here. Likewise,

the SVM-Agent also learns to cap the molecules in this manner, and the highest-scoring cluster centroids also contain aliphatic chains with rings in the linker, although commonly pyrrolidine and diazepane, as opposed to piperidine or piperazine. At least one protonatable nitrogen is common across most structures (from either origin), mostly located in the aliphatic linker. Somewhat concerningly, some of the example structures shown in Figure 3.15 have the potential to be di-cationic. This can be undesirable from a drug discovery perspective due to low logD and thus, potential implications with high clearance and low permeability. Upon further investigation (see Figure B.4) the distribution of formal charge for the Glide-Agent closely resembles that of known DRD2 actives which was predominantly +1. In fact, the SVM-Agent is slightly shifted towards containing more di-cationic molecules (~30%), despite the SVM being trained on known DRD2 actives (<10% di-cationic). Furthermore, the Glide-Agent was able to improve the docking score distribution from the Prior for all formal charges (see Figure B.5). Overall, no evidence that di-cationic molecules were preferred by the Glide-Agent due to any biases in the scoring method was found.

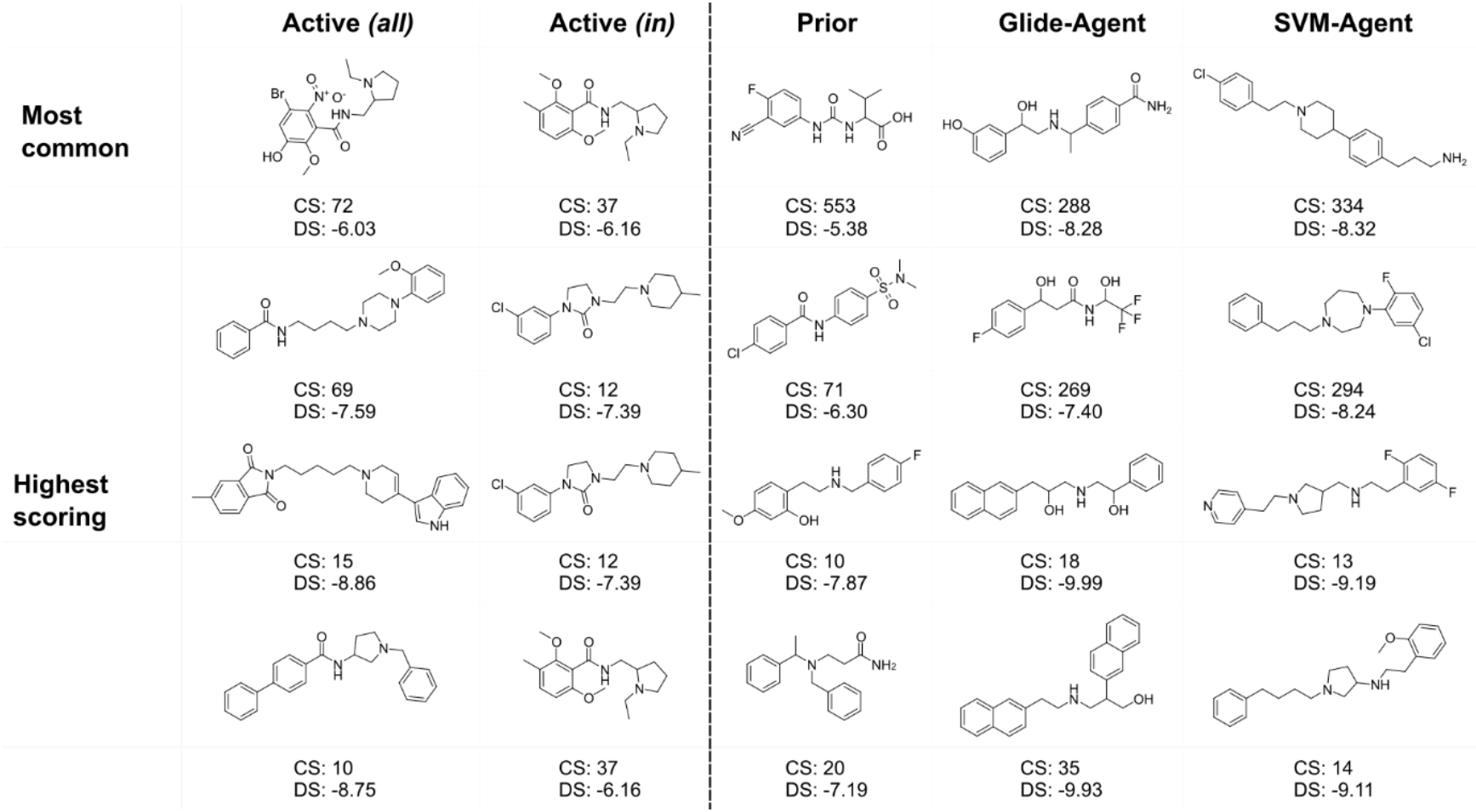


Figure 3.15: Most common and highest-scoring chemotypes of two most highly populated and the two highest-scoring clusters for each individual dataset, annotated by cluster size (CS) and mean cluster docking score (DS). The Glide- and SVM-Agent generated molecules show similar mono- or bicyclic capping of molecules as known DRD2 active molecules.

One crucial requirement of *de novo* molecules for practical use is synthetic accessability. Here, both Prior and Agent generated molecules closely inherit the SAscore distribution of the ZINC training dataset (see Figure B.6) which is likely due to the inclusion of Prior likelihood in the optimization function[76]. Despite the fact that goal-directed optimization tasks have previously led to worse syntheizability[283]. Therefore, there is no need to add proxy functions such as SAscore or QED to the optimization function (unlike recent approaches[57,284]) due to stringent filtering of the training dataset, of which the model does not deviate too much.

3.3.7 Understanding method behaviour at the ligand-protein interaction level

In order to interpret the interactions formed by *de novo* ligands originating from the different methods also at the ligand-protein interaction level. The docked poses of the two highest-scoring and the two most common cluster centroids from Figure 3.15 were generated as shown in Figure 3.16. As expected, known DRD2 ligands form a hydrogen-bond interaction with D114^{3x32}, a highly conserved residue in aminergic receptors that has been shown to be crucial for ligand binding[267,268]. This reproduction of charge interactions with D114^{3x32} can be observed in the highest-scoring molecules across all datasets, while in this instance, the Glide-Agent molecules show more distinct D114^{3x32} interaction types (e.g. also hydroxyl interactions, Figure 3.16) and vectors.

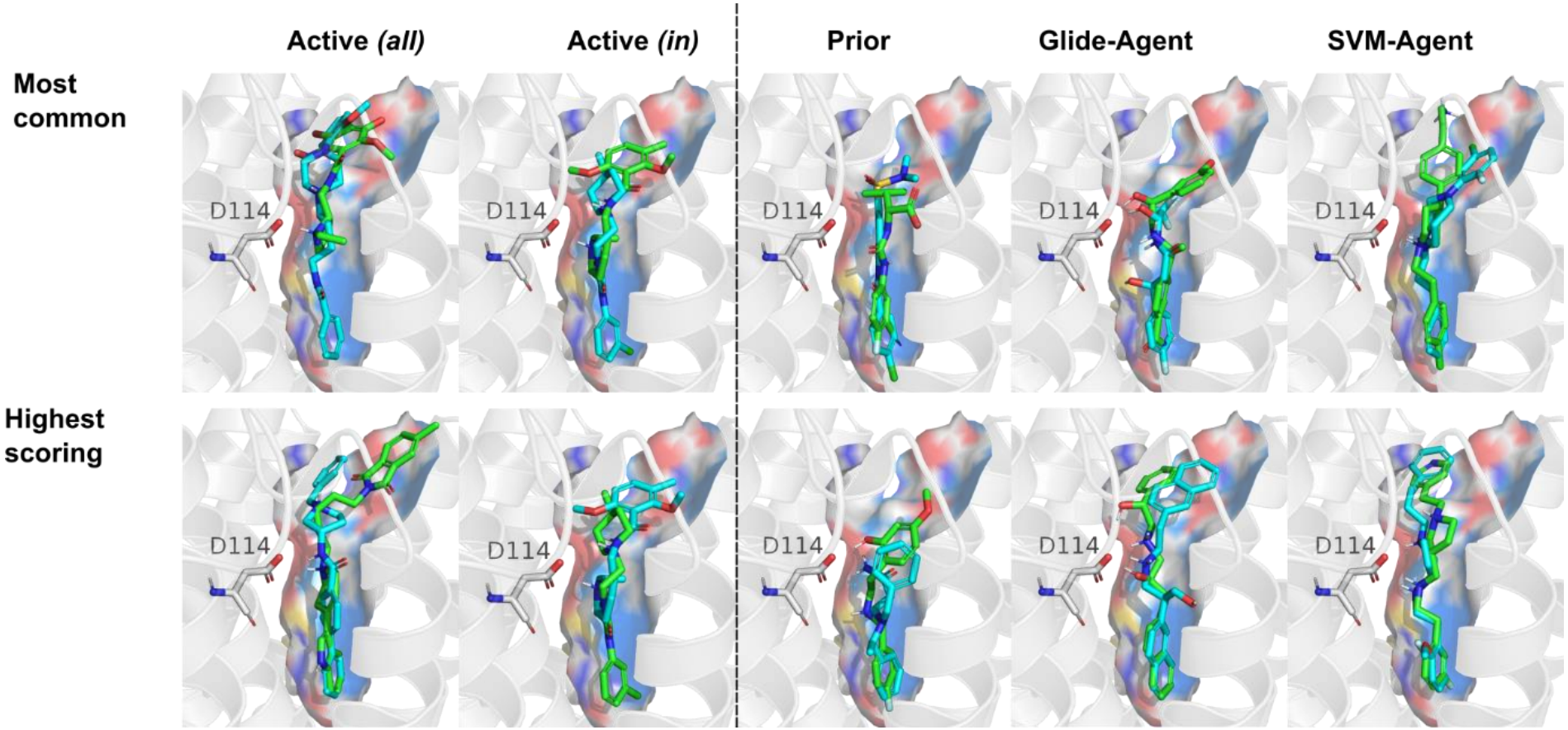


Figure 3.16: Docked pose of the cluster centroids of the two most common and highest-scoring chemotypes with DRD2. The highest-ranked ligand in both cases is displayed with sticks (green), and the second-highest ligand with lines (cyan). The Glide- and SVM-Agent examples both reproduce crucial D114^{3x32} interactions.

To understand the protein-ligand interactions present in the datasets on a broader scale, Structural Interaction Fingerprints (SIFts)[279] were calculated. Figure 3.17 summarises the changes in these interactions observed relative to the Prior (as a baseline) visually. All DRD2 binders extracted from ExCAPE-DB tend to form more interactions with residues located higher in the pocket (towards the extracellular surface). While the Glide-Agent molecules more often satisfy interactions deeper in the pocket and less often shallower ones (dissimilar to known DRD2 active molecules). Likewise, SVM-Agent molecules more often form interactions with residues deeper in the pocket. This is likely partially due to the restriction in molecular weight imposed by the ZINC subset used to train the Prior, which selects molecules with a molecular weight between 250 and 350 Daltons, subsequently biasing *de novo* molecule generation to a similar molecular weight range. Furthermore, when only considering actives with the same filters applied (i.e., molecular weight 250-350 Da) there are few residue interaction differences compared to Prior generated molecules. Surprisingly, the Glide-Agent *de novo* molecules have a lower molecular weight distribution (see Figure B.6), showing that in the current case smaller molecules are favourable for optimizing docking score, resulting in increased virtual ligand efficiency. This is in contrast to previous publications, which frequently found that larger molecules are favoured by many scoring functions[285,286]. Although there is no relative change in the sum of interactions satisfied with D114^{3x32} (despite its crucial role in ligand binding), the ratio of interaction type changes between datasets. The Glide-Agent *de novo* dataset has a higher fraction of charged hydrogen-bonding interactions (~0.75) than the Prior (~0.4), SVM-Agent (~0.6) and known DRD2 actives (~0.4-.5), where all other interactions are comprised of charged non-hydrogen-bonding interactions (see Figure 3.18). In addition, charged hydrogen-bonding interactions were associated with a better docking score distribution than charged non-hydrogen-bonding interactions (see Figure 3.19), an association which is also experimentally confirmed with higher affinity[287]. In summary, Glide-Agent optimized *de novo* molecules satisfy more charged hydrogen-bonding interactions with D114^{3x32} and generate lower molecular weight molecules than known DRD2 active molecules and SVM-Agent *de novo* molecules.

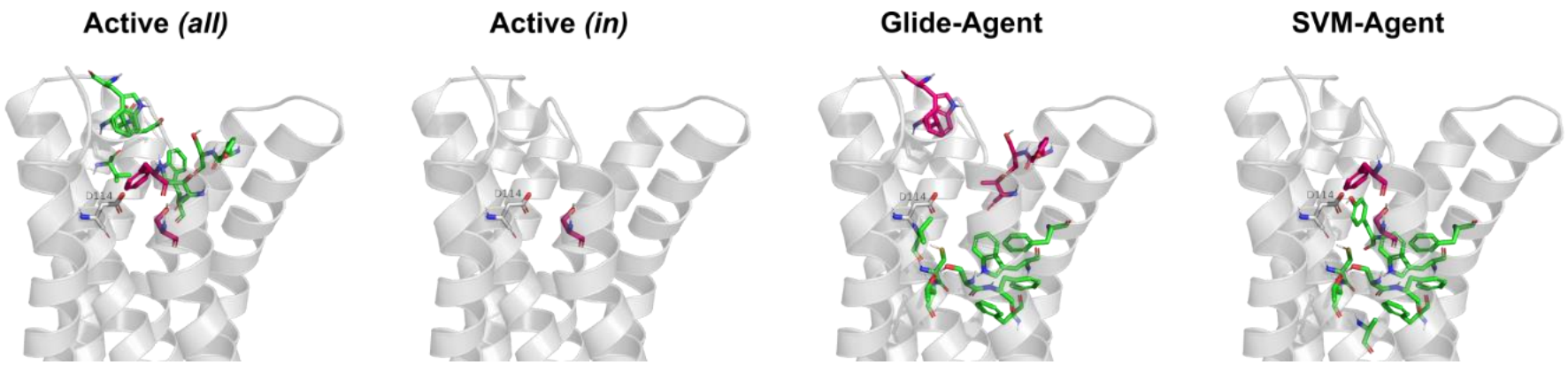


Figure 3.17: Change in the frequency of DRD2 residue interactions relative to Prior *de novo* molecules according to Structural Interaction Fingerprints (SIFTs). Green indicates a relative increase equal to or more than 10% than Prior molecules, while red indicates a decrease less than or equal to 10%.

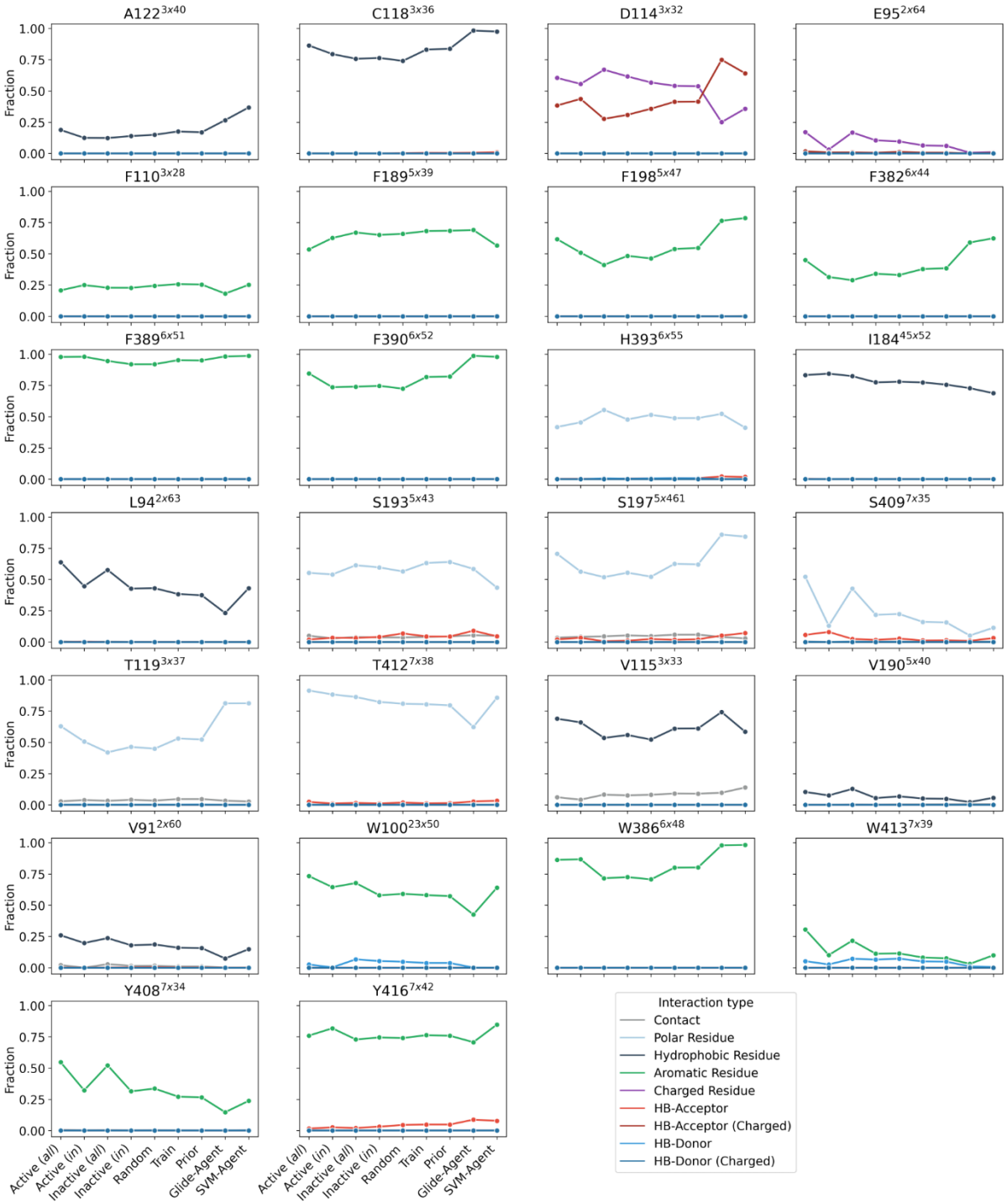


Figure 3.18: Fraction of SIFts satisfied by molecules analysed according to interaction type. The ratio of interaction types against D114^{3x32} switches for the Glide- and SVM-Agent *de novo* molecules, although more so for the Glide-Agent molecules.

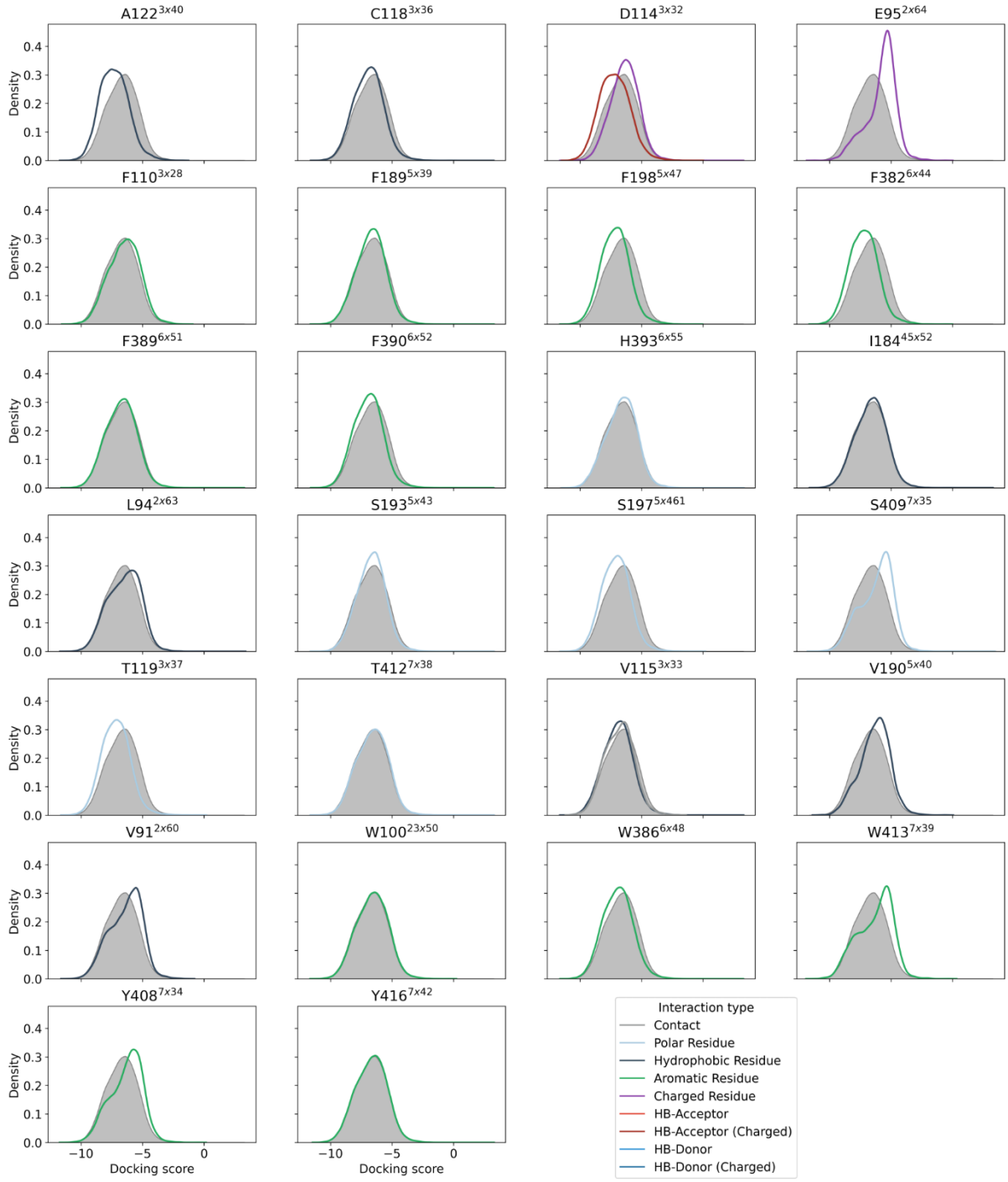


Figure 3.19: Association of residue interactions with docking score. Kernel density estimates of all docking scores (grey, shaded) and docking scores only when respective interactions are satisfied (coloured, unshaded). Of note, the D114^{3x32} HB-Acceptor (Charged) interaction is associated with better docking scores than Charged Residue interaction.

3.4 Conclusion

In this chapter, ligand-protein docking was integrated with an RNN-based generative molecular *de novo* design algorithm and compared to a ligand-based scoring function. On a benchmark dataset for the Dopamine D2 receptor this approach results in chemically sensible molecules, which can improve docking scores beyond that of known receptor ligands, while exhibiting increased physicochemical diversity compared to using the ligand-based scoring function. The work presented here demonstrates the use of deep generative models in settings also where no ligand data is available, or novelty is of particular interest (provided an x-ray crystal structure or a suitable homology model of the target is available). Moreover, this work only investigates the optimization of the Glide docking score and does not validate alternative structure-based scoring functions. While other scoring functions are expected to be equally optimizable, the resulting *de novo* chemistry may differ as a function of other forcefield implementations and/or scoring function definitions such as changes in interaction terms, for example. Overall, the ability to optimize docking score opens up the use of *de novo* design to more realistic drug discovery scenarios and takes advantage of the benefits of using structural knowledge.

Chapter 4: Augmented Hill-Climb increases reinforcement learning efficiency for language-based de novo molecule generation

4.1 Introduction

The use of MolScore presented in Chapter 2 in combination with a SMILES-based RNN with RL (REINVENT[76]) in Chapter 3 contributes towards the growing body of evidence[104,288,289] in the potential of RNNs for *de novo* drug design. Furthermore, they still match the state-of-the-art on several *de novo* molecule generation benchmarks[54,79,80,124], as discussed in 1.2.3.2. In particular, RL can be applied in more relevant drug discovery scenarios which is limited by the accuracy and reliability of scoring functions used and their relevance to the respective objective[134,135], as opposed to data availability in orthogonal fine-tuning approaches. Moreover, in Chapter 3 it was shown that RL can optimize for more complex scoring functions such as molecular docking, in contrast to some literature reports[80]. Several RL strategies have been combined with RNNs including Hill-Climb (HC)[54,122], REINFORCE[290] (used by Popova et al.[77]) and REINVENT[76], as discussed in 1.2.2.5. 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[54,80,124].

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, as observed in Chapter 3. In other examples, 163,840 molecules were sampled during HC optimization for GuacaMol benchmark tasks[54] and 192,000 molecules were sampled during REINVENT optimization of DRD2 predicted activity[76] (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[57,58,115,126,146–148,291,292]. As Chapter 3 showed that this approach can result in more diverse and novel compounds with a broader coverage of known active space than an equivalent QSAR model trained on known ligands[147]. Other studies have used

ML to build a surrogate model to predict docking scores or other physics-based scoring functions which is less computationally expensive[80,260,293]. However, the model of a model approach lessons the ability to extrapolate novel chemical space by imposing an extra applicability domain, and compounds prediction uncertainty on top of pre-existing inaccuracies[141,178]. Therefore, it is attractive to improve the sample efficiency of RL optimization to enable routine use of such computationally expensive scoring functions directly.

Previous work has explored RL strategies and parameters for RNNs *de novo* molecule generation to varying degrees. Niel *et al.*[122] 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 or investigated in the publication[99]. Fialková *et al.*[294] investigated more significant modifications to the REINVENT loss function which did not result in any significant improvement. Meanwhile, Atance *et al.*[123] 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, a very simple change to the REINVENT strategy is made to ameliorate overpowered regularization by introducing elements of the HC strategy. I call this novel hybrid approach Augmented Hill-Climb (AHC) and investigate it’s use for RNN *de novo* molecule generation. I further compare AHC to previously mentioned RL strategies that are implemented in published studies.

4.2 Materials and methods

The evaluation of AHC and comparison to other RL strategies was built around four key experiments which are summarised in Figure 4.1 (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[54]. Experiment 4, benchmark comparison between AHC and other RL strategies on six tasks of varying difficulty. Experiment 5, benchmark comparison between AHC and REINVENT on alternative language-based generative models (a transformer architecture and reinforcement learning stabilized transformer architecture) on the same benchmark tasks as Experiment 4.

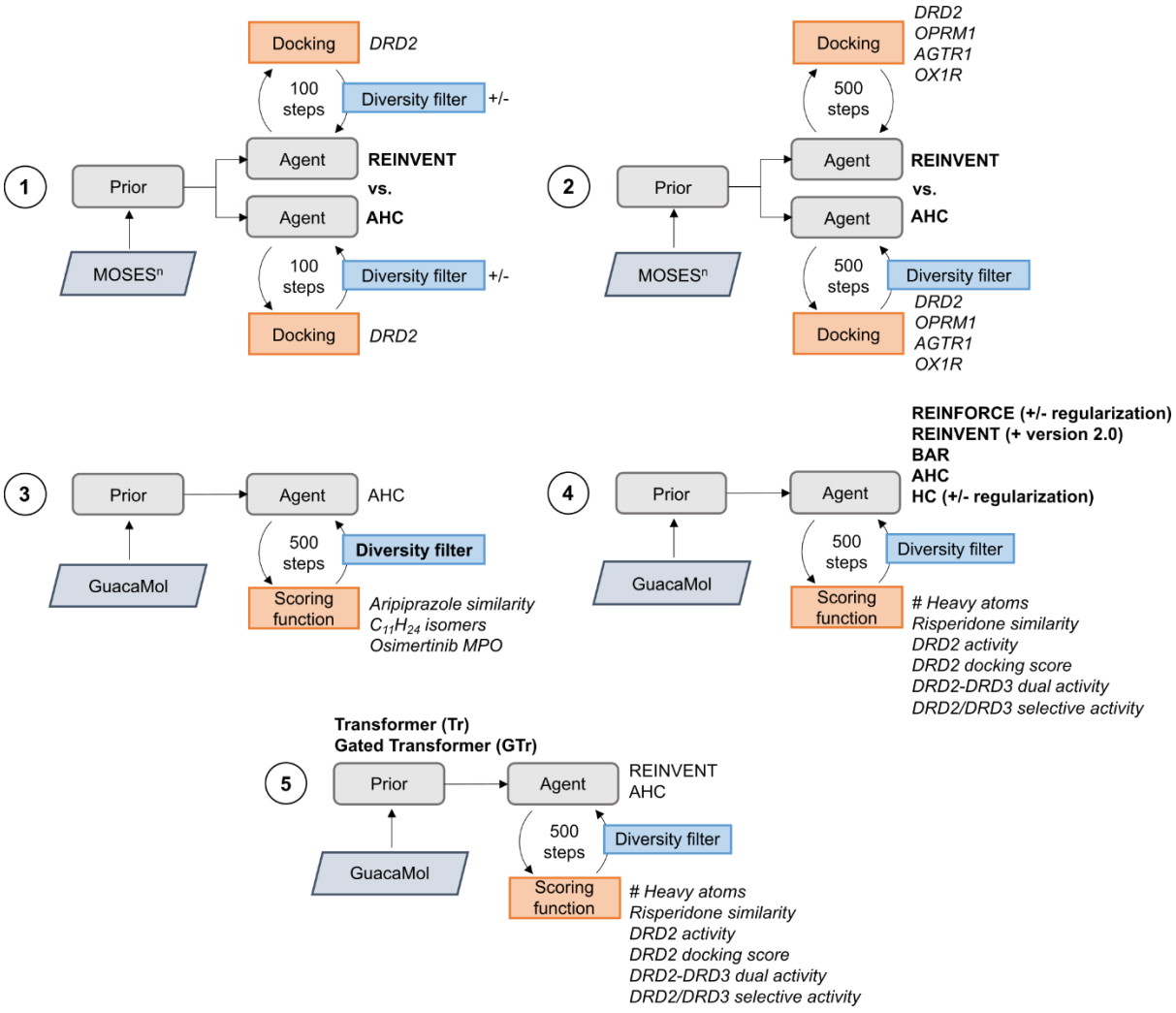


Figure 4.1: Schematic of the five experiments conducted in this chapter with the focus of each experiment in bold face. In each case the Prior and Agent refer to an RNN. 1) Comparison of AHC to REINVENT on a single docking task over 100 RL updates. 2) Comparison of AHC to REINVENT on four different docking 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. 5) Comparison of AHC to REINVENT on two different transformer architectures on six benchmark tasks.

4.2.1 Training data

RNNs were trained using either a modification of the MOSES dataset or the GuacaMol dataset. The MOSES^{n} dataset of size 2,454,048 as prepared in 3.2.1. The GuacaMol train dataset[54] (1,273,104 molecules) derived from ChEMBL24 containing real molecules both in the 'drug-like' domain and others such as peptides and natural products. The GuacaMol dataset applies the following filters during curation: salt removal; charge neutralization; molecules with SMILES strings shorter than 100 characters; 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 than MOSES^{n}.

4.2.2 Recurrent neural network

The RNN implemented in this work is the same as[29,76,99,295]. 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[76]. 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[99]. The third configuration consisted of three LSTM layers of size 512 with a dropout rate of 0.2, consistent with the GuacaMol implementation[54] as found on the corresponding GitHub repository[296]. 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.

4.2.3 Transformer

Two transformer encoder architectures were used in this work. The first is the original proposed Transformer (Tr) encoder[84] as described in 1.2.2.4. The second is a Gated Transformer (GTrXL) encoder adapted from Parisotto et al.[297] in an attempt to stabilize

the Transformer under RL conditions. Figure 4.2 shows that the key differences in this second architecture: 1) relocation of layer normalization before the respective sub-layer (i.e., multi-head attention or feed-forward network), and 2) employment of an adapted GRU gating mechanism in place of the residual connection shown in Equation 4.1. Notably the original adaption was applied to Transformer-XL[298] which contains a memory mechanism to expand context for larger language tasks. This memory mechanism was omitted for simplicity and due to the shorter nature of SMILES strings which are typically 20-100 characters long compared to 10^3-10^4 in large language tasks. Therefore, this model is simply referred to as Gated Transformer (GTr).

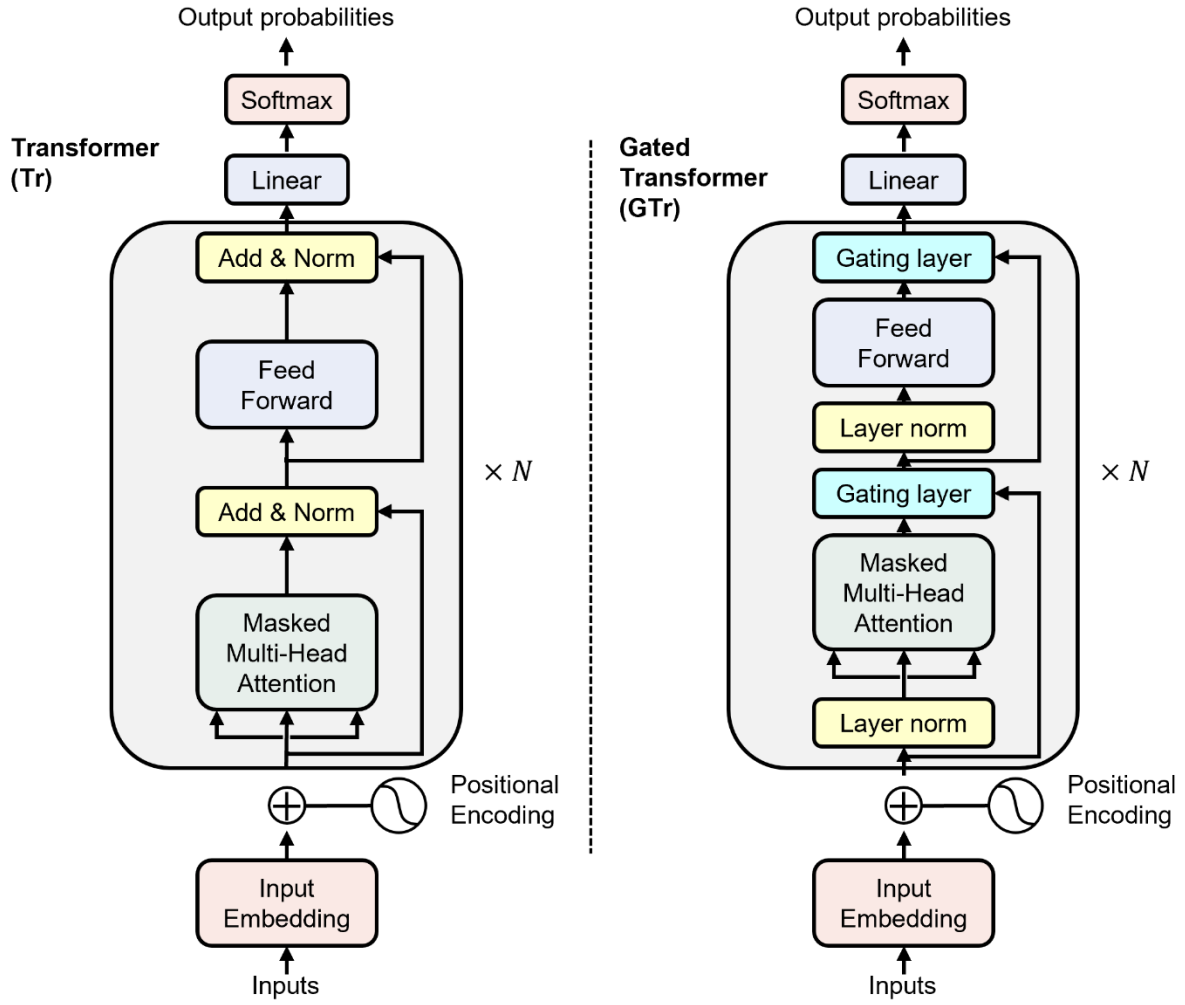


Figure 4.2: Comparison of the encoder-only Transformer architecture for AR sequence generation *versus* the Gated Transformer architecture proposed by Parisotto et al.[297] to stabilise learning under RL conditions.

$$
\begin{aligned}
r &= \sigma(W_r y + U_r x) \\
z &= \sigma(W_z y + U_z x - b_g) \\
\hat{h} &= tanh(W_g y + U_g(r \odot x)) \\
g(x, y) &= (1 - z) \odot x + z \odot \hat{h}
\end{aligned}
$$

Equation 4.1

The hyperparameters were the same for both the architectures. More specifically, each consisted of 4 encoder layers with hidden dimension 512, each with 8 multi-attention heads and finally a feed-forward network of hidden dimension 1,024. A dropout of 0.1 was used throughout. Each model was then trained on the GuacaMol training dataset for 5 epochs with a batch size of 128 and the ADAM optimizer with a learning rate of 0.001.

4.2.4 Augmented Hill-Climb

Here, Augmented Hill-Climb (AHC) is proposed which can be viewed as a hybrid strategy between HC[122] and REINVENT[76], as depicted in Figure 4.3. With AHC the loss is calculated as in REINVENT (by defining the augmented likelihood) but only on the top k molecules, 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 ($R_T \rightarrow 0$) are sampled the score contribution goes to zero and $logP_{\mathbb{U}}(A) \approx logP_{prior}(A)$. In this situation the agent policy will, in-fact, trend back towards the prior policy which may negate useful learnings, as the loss function is effectively a distance (see Equation 1.10). Low scoring molecules 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, ensuring prior learnings are not ‘forgotten’.

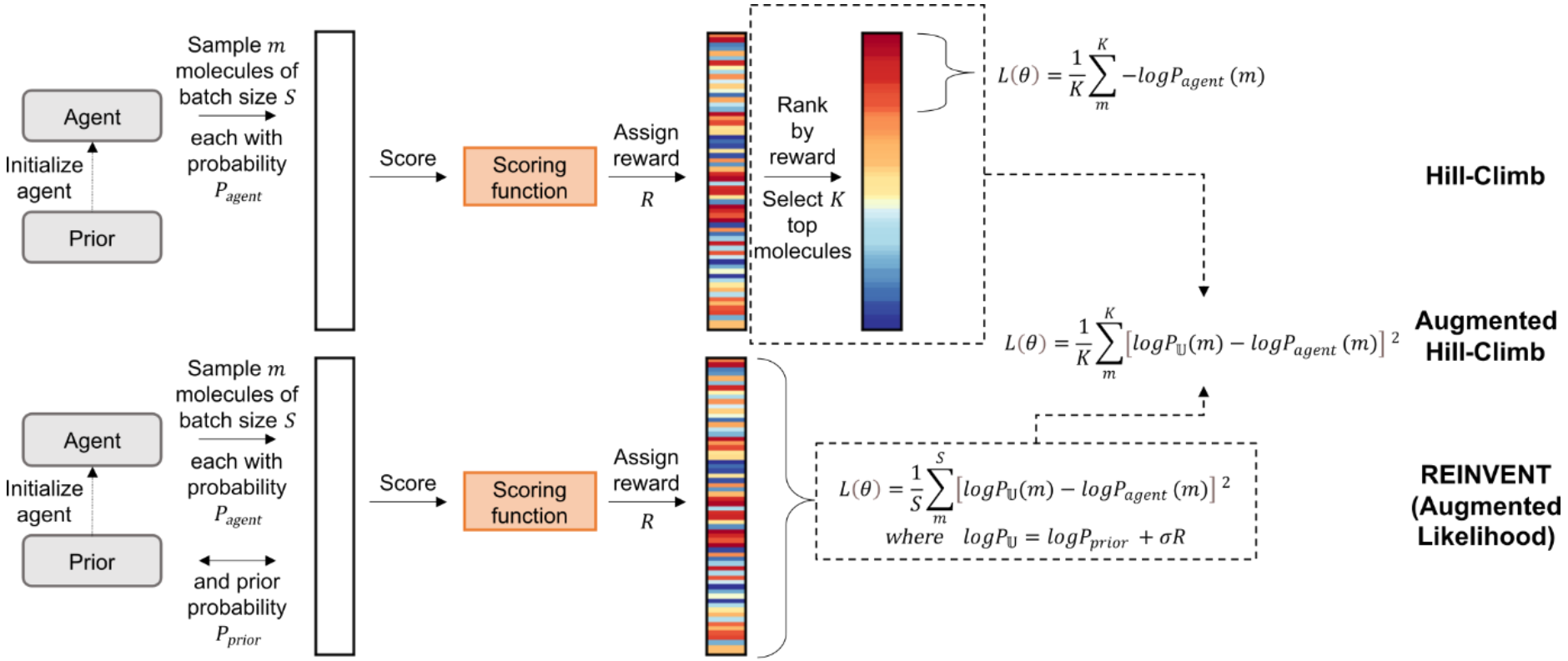


Figure 4.3: Depiction of the REINVENT, Hill-Climb (HC) and Augmented Hill-Climb (AHC) optimization algorithms and subsequent loss functions $\mathcal{L}$ as parameterized by network parameters θ. AHC is a hybrid algorithm that combines elements of REINVENT and HC.

4.2.5 Reinforcement learning

Aside from Augmented Hill-Climb, all RL baselines used in this chapter are described in 1.2.2.5, further implementation hyperparameters are shown in Table 4.1.

Table 4.1: Default hyperparameters used for reinforcement learning strategies benchmarked in this chapter. Additional configurations with regularization and disclosed in square brackets. Separated at the bottom are configurations used in the PMO benchmark.

RL strategy	Training steps	Batch size	σ	k	λ_{KL}	lr	α
REINFORCE [+reg]	500	64	-	-	[10]	1×10^{-4}	-
REINVENT	500	64	60	-	-	5×10^{-4}	-
REINVENT 2.0	250	128	120	-	-	5×10^{-4}	-
BAR	500	64	60	-	-	5×10^{-4}	0.5
Augmented Hill-Climb	500	64	60	50%	-	5×10^{-4}	-
Hill-Climb [+reg]	32	1024	-	50%	[10]	5×10^{-4}	-
Hill-Climb* [+reg]	500	64	-	50%	[10]	5×10^{-4}	-
SMILES-AHC	40[a]	256	120	25%	-	5×10^{-4}	-
SMILES-AHC*	40[a]	256	60	25%	-	5×10^{-4}	-

[a] Optimization was capped at 10,000 molecules

4.2.6 Diversity filters

The DFs (see 2.2.1.4) and parameters used in this work (i.e., DF1, DF2 and DF3) are shown in Table 4.2. DFs parameters are further explored in Experiment 3.

Table 4.2: Diversity filter configurations used in this chapter.

	DF1	DF2	DF3
Topology simplification	BM scaffold	BM scaffold	BM scaffold
Fingerprint type	ECFP (Morgan)	ECFP (Morgan)	ECFP (Morgan)
Fingerprint radius	2	2	2
Fingerprint bits	1024	1024	1024
Similarity metric	Tanimoto	Tanimoto	Tanimoto
Minimum score threshold	0.8	0.5	0.0
Bin size	25	50	50
Output mode	Binary	Linear	Linear

4.2.7 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 4.3 and are described in more detail in the subsequent sections. All scoring functions were implemented using the MolScore platform[150] as described in Chapter 2.

Table 4.3: Summary of all objectives / tasks used in this chapter, and for which experiment (see Figure 4.1).

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

4.2.7.1 Target preparation and docking tasks

Four different targets were used for molecular docking scoring functions to evaluate docking score optimization (Experiments 1, 2 and 4 in Figure 4.1). The four targets and corresponding x-ray crystal structures used in the docking tasks were D_2 (DRD2, PDB: 6CM4[259]), μ (OPRM1, PDB: 4DKL[299]), AT_1 (AGTR1, PDB: 4YAY[300]) and OX_1 (OX1R, PDB: 6TO7[301]) receptors.

All target crystal structures were first prepared using Schrodinger Protein Preparation Wizard[302] using default parameters which included: addition of protein and ligand hydrogens (pH 7±2, Epik[219]), optimization of hydrogen bond networks (pH 7, PROPKA[272]), restrained minimization using the OPLS3e force field[273], 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)[301].

Docking was conducted using Glide-SP[212] as described in 2.2.1.1 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[262]. 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 C.1. Classification accuracy, precision and recall were assessed by varying docking score decision thresholds (Figure 4.4). 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%.

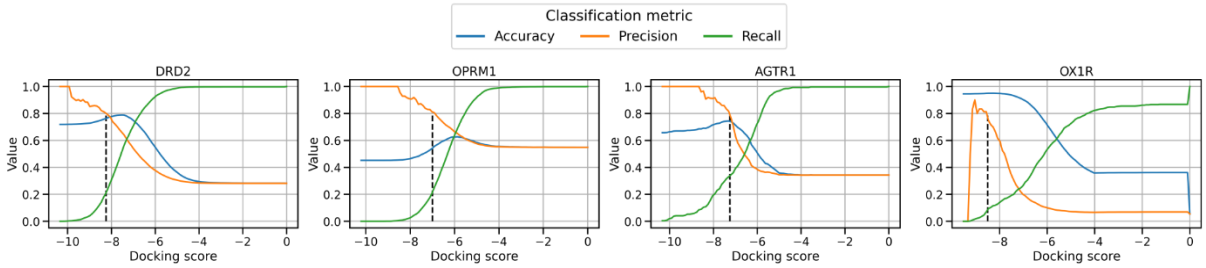


Figure 4.4: Retrospective classification performance of docking protocol on the four targets investigated in this chapter. In each case, the accuracy, precision, and recall are traced with varying docking score decision thresholds. Thresholds corresponding to ~ 80% precision are annotated by black dashed lines.

4.2.7.2 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 tasks from the GuacaMol benchmarking suite[54] were chosen and re-implemented according to the original work: Aripiprazole similarity, $C_{11}H_{24}$ isomers and Osimertinib MPO. The goal of 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 (molecules are penalized if too close) 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 and symptom of mode collapse), goal (the score returned by the scoring function/s) and run time (a practical measure to identify if some DFs are slower to compute).

4.2.7.3 QSAR model training

Active and inactive molecules against DRD2 and against DRD3 were extracted from the ExCAPE-DB[262]. 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[303] on ECFP4 fingerprints with 2,048 bits (default bit length, as 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[203]. 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[304] resulting in the performance shown in Figure 4.5.

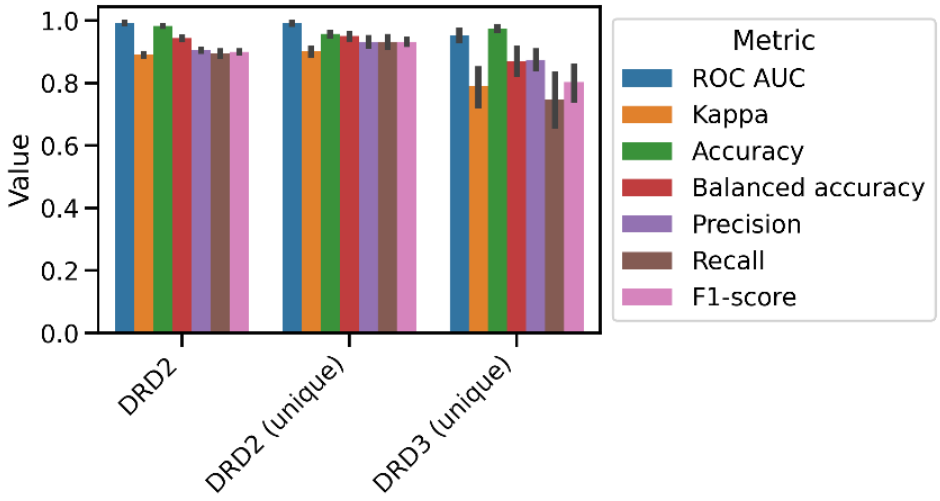


Figure 4.5: QSAR model performance of RF models trained on DRD2 and DRD3 active and inactive molecules, based on 5-fold stratified, clustered cross-validation with GHOST[304] decision threshold identification.

4.2.7.4 DRD2 benchmark tasks

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

 # Heavy atoms – This 'easy' task aims to maximize the number of heavy atoms in a molecule calculated by RDKit[199]. This is similar in concept to maximizing penalized LogP[106] and QED[173] which has been shown to be trivial by some generative models[106,117,305]. This predominantly probes the RL strategy's ability to extrapolate beyond the training dataset (which contains molecules with a limited number of heavy atoms), rather than being a measure of good performance. However, this task is irrelevant to real drug discovery objectives.

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 used as default for similarity measurements in MOSES[79] (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. The ability of generative models to easily maximize such tasks has been shown in benchmark studies[54].

DRD2 activity – This 'medium' task aims to maximize the QSAR predicted probability of activity against DRD2 (Equation 4.2). This task is representative of a real objective during early-stage hit finding, providing that known ligand data is available. Maximization of these tasks are often achieved by generative models[76,77,117] but it is a more scientifically complex objective than molecular similarity.

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

Equation 4.2

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[199] and then conducting protonation using Epik (pH 7.4) to only protonate the most abundant state per stereoisomer. This task has been successfully optimized by generative models in some cases[57] but proven difficult in others[80].

DRD2-DRD3 dual – This 'hard' task aims to maximize the QSAR predicted probability of activity against both DRD2 and DRD3 (Equation 4.3). This task is representative of real drug discovery projects requiring polypharmacological activity, providing that ligand data for both is available. This constitutes a multi-objective optimization problem which has proven more difficult for generative models with an increasing number of constraints[57,117,306].

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

Equation 4.3

DRD2/DRD3 selective – This 'hard' task aims to maximize the QSAR predicted probability of selective activity against DRD2 over DRD3 (Equation 4.4). 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. Similar to dual inhibition, multi-objective optimization problems are more difficult for generative models to optimize against[117,306].

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

Equation 4.4

4.2.8 Practical molecular optimization benchmark measures

For endpoint measurement in the practical molecular optimization benchmark, AUC of the ten highest-ranking molecules generated during optimization is retained to account for sample efficiency (AUC Top-10)[55]. However, I additionally modify how these molecules are identified.

AUC Top-10 (Filtered) To include a property constraint to the initial training data, this metric filters out molecules that have a molecular weight or LogP beyond 4 standard deviations from the mean of pre-training dataset ZINC250k ($\mu \pm 4\sigma$ approximately contains 99.99% of a normal distribution). As a measure for topological idiosyncrasies, I additionally filter out molecules that contain more than 10% *de novo* (unobserved in ZINC250k) ECFP4 fingerprint features (implemented in RDKit). The 10% threshold was chosen by inspecting molecules with varying fractions of *de novo* ECFP4 features and therefore there is scope for further justification of this threshold. These simple and lenient filters should be satisfied as a minimum requirement to ensure that the generative model does not drift beyond its applicability domain (if the model is distribution-based), or at-least maintains some similarity to the training dataset on the basis that the dataset contains practically relevant chemistry.

AUC Top-10 (Diverse) Gao et al. rationalise the selection of the highest-ranking ten molecules as "distinct molecular candidates to progress to later stages of development". I explicitly enforce this by selecting ten diverse molecules iteratively, where a molecule is only added to the selection if its Tanimoto similarity to any previously selected compounds is not higher than 0.35 (by ECFP4 fingerprints with 1,024 fingerprints, implemented in RDKit). Anything more similar than threshold with

these fingerprint parameters broadly correlates to an 80-85% probability of belonging to the same bioactivity class[307], but distinct candidates should ideally possess different profiles.

AUC Top-10 (Combined) A combination of applying both property filters and diversity filters as described above.

4.3 Results and discussion

4.3.1 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 4.1). The REINVENT strategy and docking protocol was identical to Chapter 2.

To increase optimization power, the easiest and most obvious change is to increase the score contribution to the augmented likelihood used by REINVENT by increasing the scalar value σ. The original work[76] had a default value of 60, however, the subsequent update (REINVENT 2.0[99]) increased this value to 120 - suggesting that sample efficiency was sub-optimal. Therefore, I 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 4.6a, this made little improvement in the optimization of DRD2 docking scores. 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 and using the same amount of RL updates (Figure 4.6a). 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, this led to 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 4.6b) i.e., AHC was more prone to mode collapse.

To address the mode collapse, a diversity filter[167] (DF1) 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 (Figure 4.6c,d). 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, it also decreases the prior contribution and thus decreases regularization during optimization. As such, larger values of σ are expected to result in further extrapolation outside the domain of the training set and prior. Figure 4.6e-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[285]). 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.

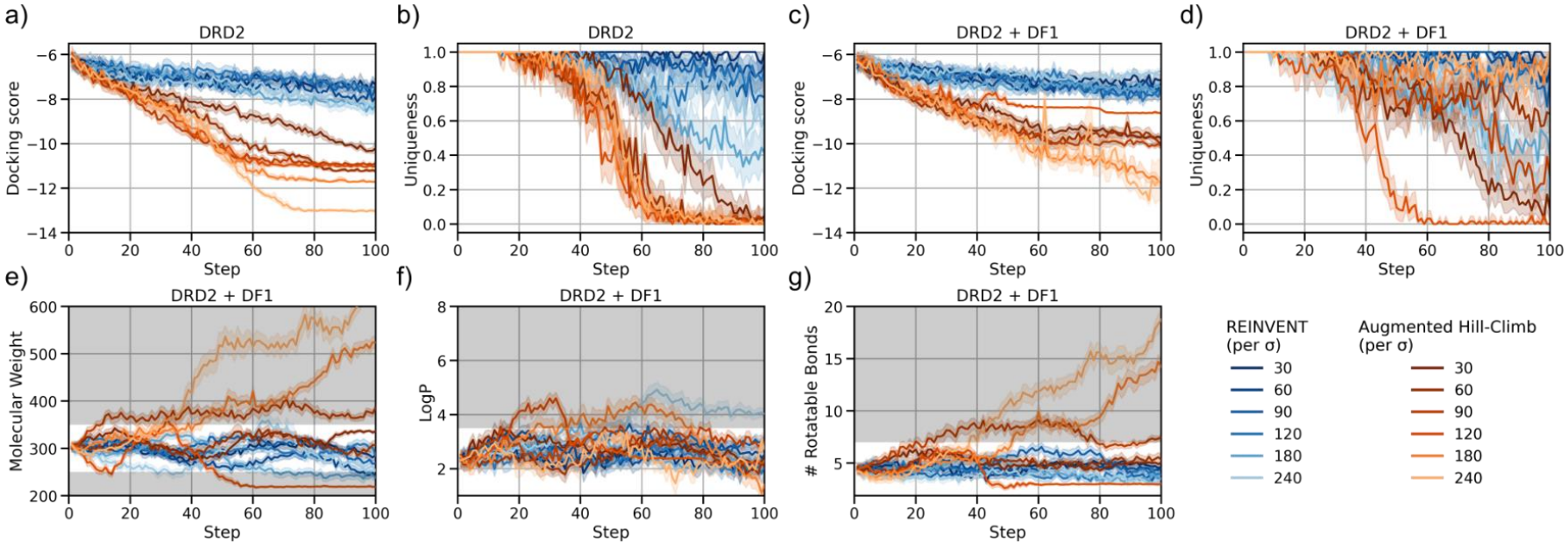
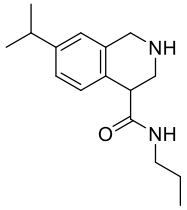
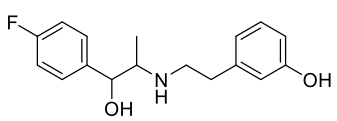
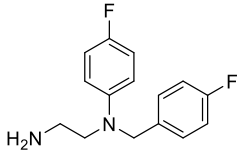
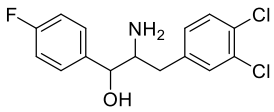
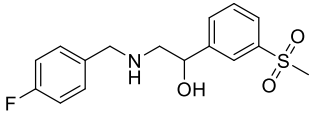
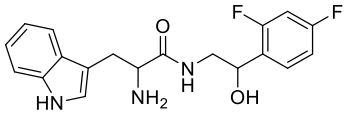
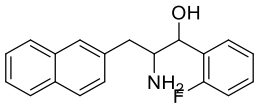
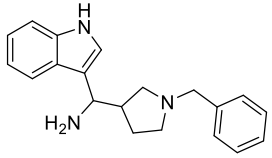
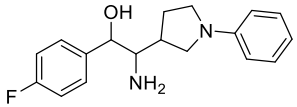
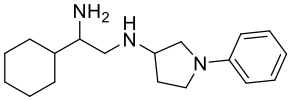
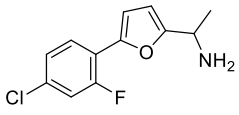
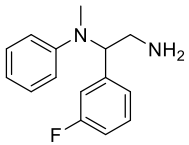
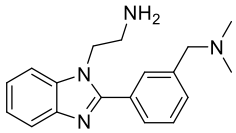
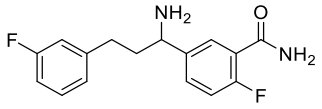
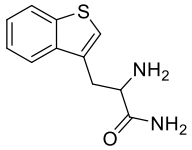
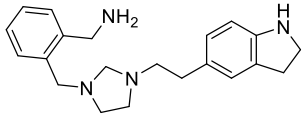
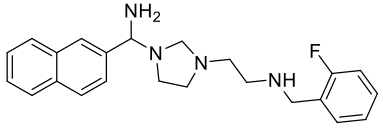
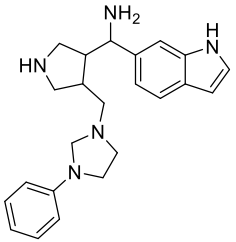
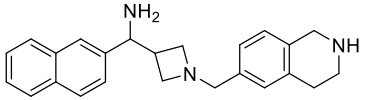
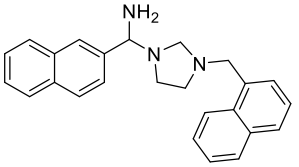
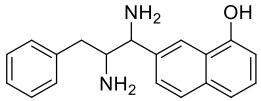
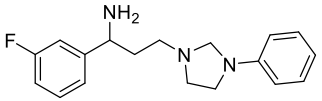
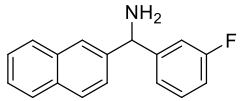
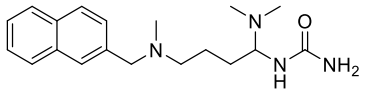
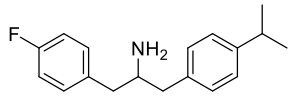
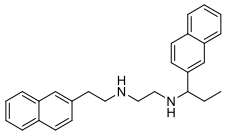
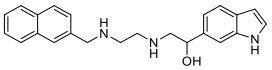
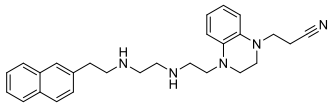
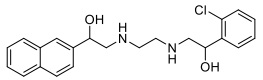
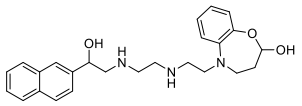
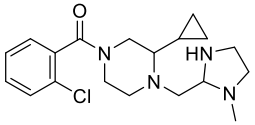
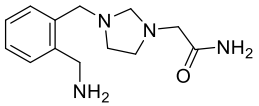
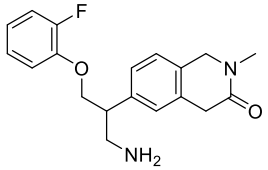
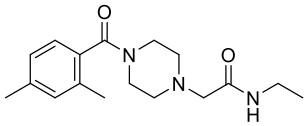
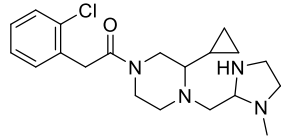


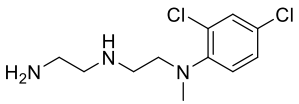
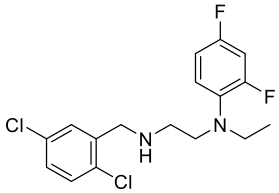
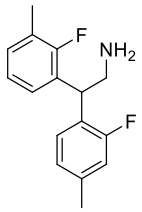
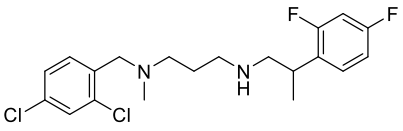
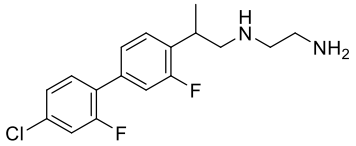
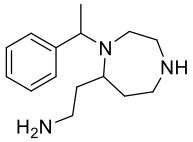
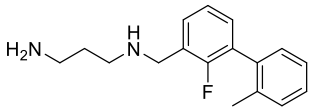
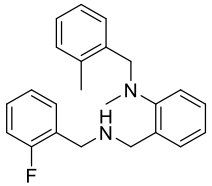
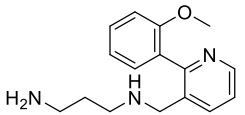
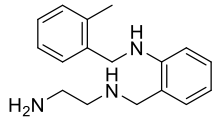
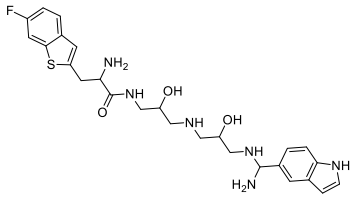
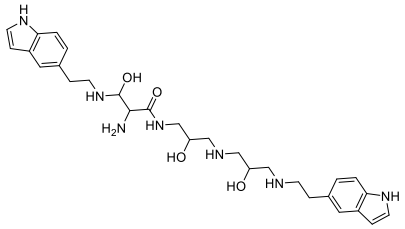
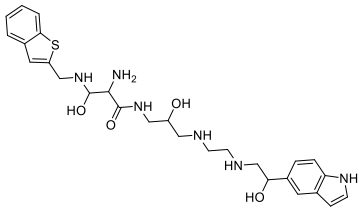
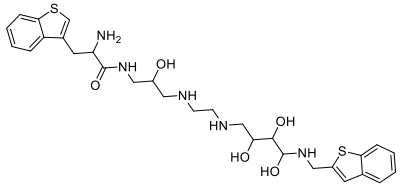
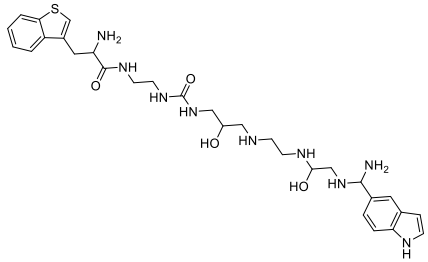
Figure 4.6: Comparison between REINVENT and Augmented Hill-Climb learning strategies to optimize DRD2 docking scores at varying levels of σ. (a) Augmented Hill-Climb is more efficient at optimizing docking score at all levels of σ but (b) undergoes increased mode collapse via a drop in uniqueness. (c) 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.

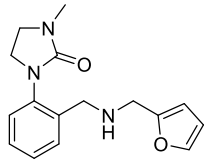
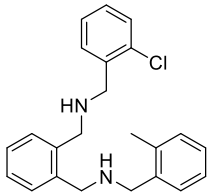
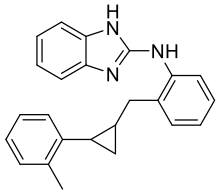
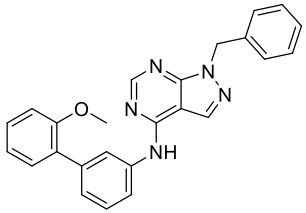
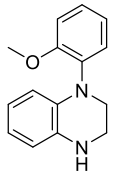
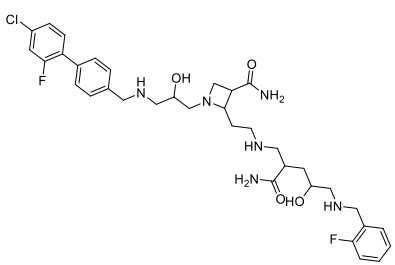
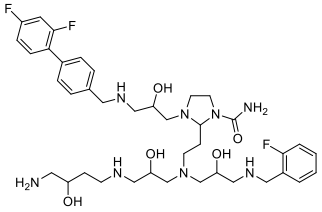
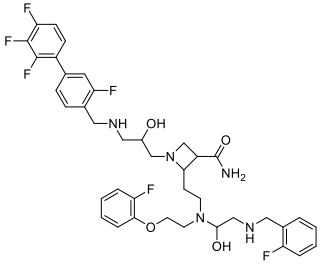
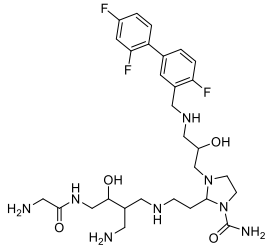
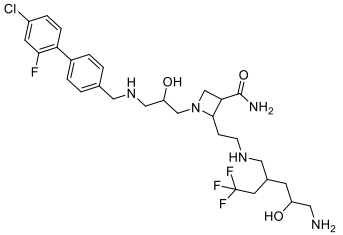
Despite improvement in the optimization ability by AHC, it is irrelevant if the resulting *de novo* structures are invalid or implausible (e.g., incorrect valences, unstable or idiosyncratic functional groups or strained ring systems). The chemistry generated by RNNs has been evaluated previously[29,34,79,104,165] 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. On the other hand, the REINVENT strategy has been shown to maintain similar chemistry to the prior RNN[76,147,148,167]. Therefore, some of the top molecules generated at different values of σ are visualized in Table 4.4. 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 in close proximity. This chemotype is consistent with the co-crystallised inverse agonist risperidone[259] and required interactions to D114^{3x32} for ligand activity[268,287,308], where the cyclic moiety sits deep in the hydrophobic sub-pocket and the cationic amine forms 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[285]. This corroborates the observation of extrapolation into restricted property space seen in Figure 4.6e,g, an ability which enables this exploitation. In this scenario additional constraints are necessary, such as also defining a suitable molecular weight range as this knowledge is no longer imposed by the prior dataset. These results highlight the balance that is required in the trade-off between regularization and optimization, which is better achieved by AHC than REINVENT.

Table 4.4: Centroid of the top 5 largest clusters for the top 100 molecules generated by REINVENT and Augmented Hill-Climb (both with DF1) according to docking score against DRD2 receptor for varying values of sigma (σ). Cluster size (CS), centroid docking score (DS) and the average cluster docking score (AvDS) is annotated below.

σ	RL strategy	Top 1	Top 2	Top 3	Top 4	Top 5
30	REINVENT	 CS: 5 DS: -9.13 AvDS: -9.53	 CS: 4 DS: -9.16 AvDS: -9.26	 CS: 4 DS: -9.28 AvDS: -9.79	 CS: 3 DS: -9.30 AvDS: -9.41	 CS: 2 DS: -9.10 AvDS: -9.36
30	AHC	 CS: 14 DS: -10.36 AvDS: -10.60	 CS: 7 DS: -10.34 AvDS: -10.57	 CS: 5 DS: -10.32 AvDS: -10.67	 CS: 4 DS: -10.40 AvDS: -10.50	 CS: 4 DS: -10.48 AvDS: -10.53
60	REINVENT	 CS: 8 DS: -9.10 AvDS: -9.28	 CS: 3 DS: -9.12 AvDS: -9.37	 CS: 3 DS: -9.42 AvDS: -9.53	 CS: 2 DS: -9.09 AvDS: -9.50	 CS: 2 DS: -9.12 AvDS: -9.50

60	AHC					
		CS: 19 DS: -11.48 AvDS: -11.64	CS: 5 DS: -11.43 AvDS: -11.52	CS: 3 DS: -11.16 AvDS: -11.25	CS: 3 DS: -11.28 AvDS: -11.54	CS: 3 DS: -11.43 AvDS: -11.45
90	REINVENT					
		CS: 4 DS: -9.53 AvDS: -9.78	CS: 4 DS: -9.56 AvDS: -9.84	CS: 3 DS: -9.59 AvDS: -9.78	CS: 3 DS: -9.66 AvDS: -9.87	CS: 2 DS: -9.39 AvDS: -9.54
90	AHC					
		CS: 10 DS: -11.21 AvDS: -11.38	CS: 7 DS: -11.23 AvDS: -11.51	CS: 7 DS: -11.53 AvDS: -11.91	CS: 5 DS: -11.24 AvDS: -11.36	CS: 4 DS: -11.33 AvDS: -11.38
120	REINVENT					
		CS: 5 DS: -9.95 AvDS: -9.84	CS: 4 DS: -9.30 AvDS: -9.77	CS: 2 DS: -9.28 AvDS: -9.40	CS: 2 DS: -9.30 AvDS: -9.41	CS: 2 DS: -9.34 AvDS: -9.42

120	AHC					
		CS: 21 DS: -9.28 AvDS: -9.64	CS: 9 DS: -9.35 AvDS: -9.91	CS: 7 DS: -9.28 AvDS: -9.58	CS: 6 DS: -9.84 AvDS: -10.16	CS: 5 DS: -9.28 AvDS: -9.52
180	REINVENT					
		CS: 4 DS: -9.19 AvDS: -9.26	CS: 4 DS: -9.35 AvDS: -9.57	CS: 2 DS: -9.10 AvDS: -9.15	CS: 2 DS: -9.12 AvDS: -9.25	CS: 2 DS: -9.16 AvDS: -9.64
180	AHC					
		CS: 8 DS: -12.80 AvDS: -12.94	CS: 7 DS: -12.77 AvDS: -12.83	CS: 6 DS: -12.58 AvDS: -12.84	CS: 5 DS: -12.76 AvDS: -13.03	CS: 4 DS: -12.60 AvDS: -12.94

240	REINVENT					
		CS: 3 DS: -8.75 AvDS: -8.81	CS: 3 DS: -9.04 AvDS: -9.24	CS: 2 DS: -8.74 AvDS: -8.56	CS: 2 DS: -8.77 AvDS: -8.94	CS: 2 DS: -8.80 AvDS: -8.96
240	AHC					
		CS: 28 DS: -13.46 AvDS: -13.77	CS: 19 DS: -13.85 AvDS: -13.96	CS: 5 DS: -13.43 AvDS: -13.92	CS: 5 DS: -13.75 AvDS: -13.89	CS: 4 DS: -13.48 AvDS: -13.69

4.3.2 Optimization of docking scores for multiple GPCR targets

Aside from DRD2 in Chapter 3, other GPCR targets (DRD2, OPRM1, AGTR1 and OX1R) were investigated using REINVENT to optimize docking score over the course of 3,000 RL updates, the first 500 updates of which are shown in Figure 4.7. DRD2[259] contains a deep hydrophobic sub-pocket and requires a salt bridge interaction with D114^{3x32} for ligand activity. OPRM1[299] similarly forms a salt bridge interaction via D147^{3x32} (a structurally conserved position in aminergic receptors[268,308]) but with a more open pocket than DRD2. AGTR1[300] 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[301] 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 4.7a,b), while the latter two targets' respective docking scores were more challenging and showed little minimization (Figure 4.7c,d) (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 minimize 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 4.1). For every target, AHC in combination with DF1 (Figure 4.7) 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, the 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 4.6. 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 4.8). 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 4.6a,b).

Therefore, a search of DFs and hyperparameters was needed to identify a more optimal configuration that would successfully and robustly rescue mode collapse (Experiment 3, Figure 4.1) (see next subsection). This led to the design of DF2 which differed from DF1 by having a lower minimum score threshold of 0.5 instead of 0.8, linear penalization output mode instead of binary, and larger bin size of 50 instead of 25. Using DF2 the previous experiment (Experiment 2, Figure 4.1) was conducted on the four targets as before, shown in Figure 4.7. 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 4.8b). To gain a quantitative understanding of improvement in sample efficiency, Table 4.5 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.

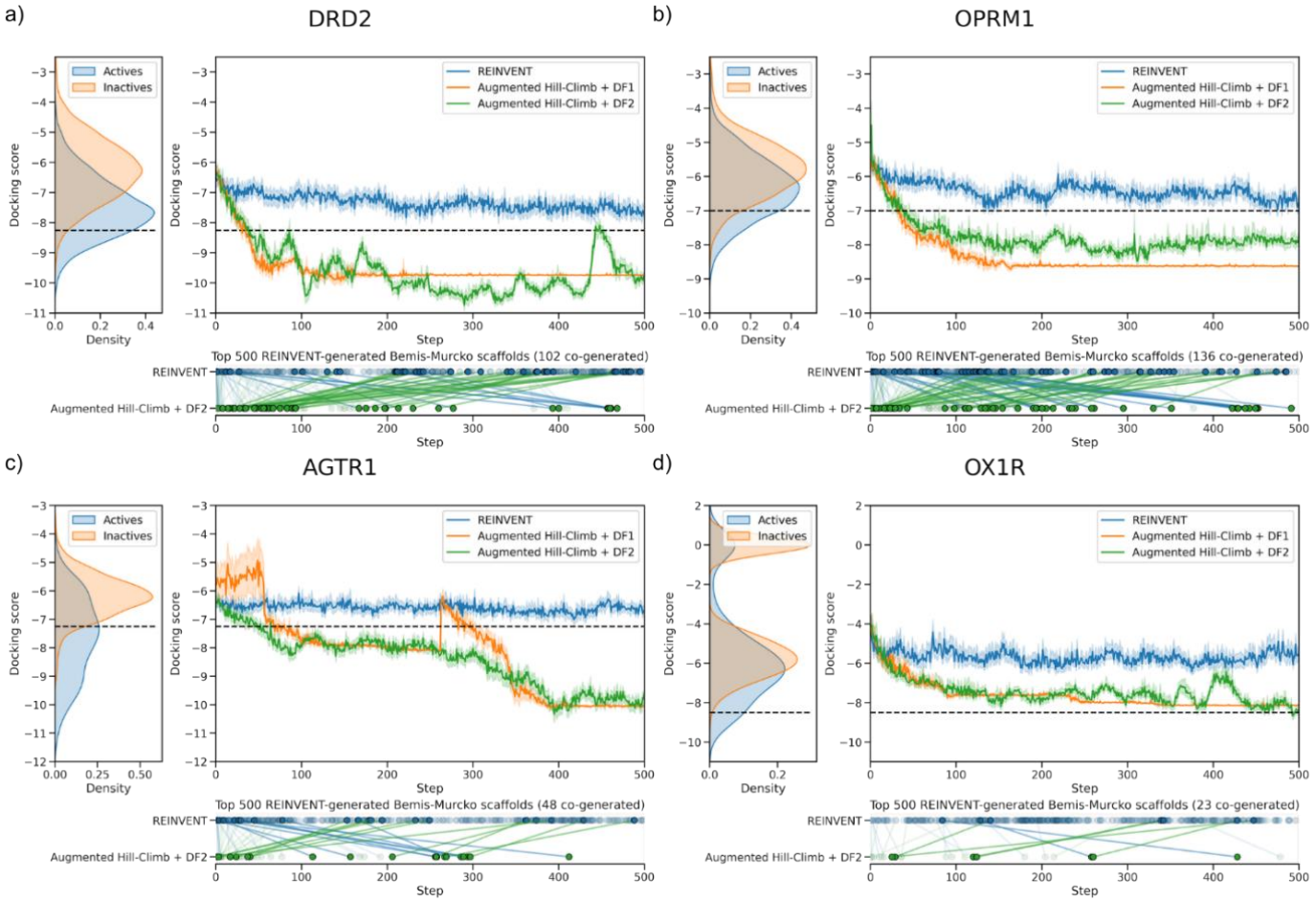


Figure 4.7: Improved learning efficiency of Augmented Hill-Climb against four targets: (a) DRD2, (b) OPRM1, (c) AGTR1 and (d) OX1R. (top left panel) Distribution of known active and inactive molecule docking scores. (top right panel) Optimization of *de novo* molecule docking score *via* RL. (bottom 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.

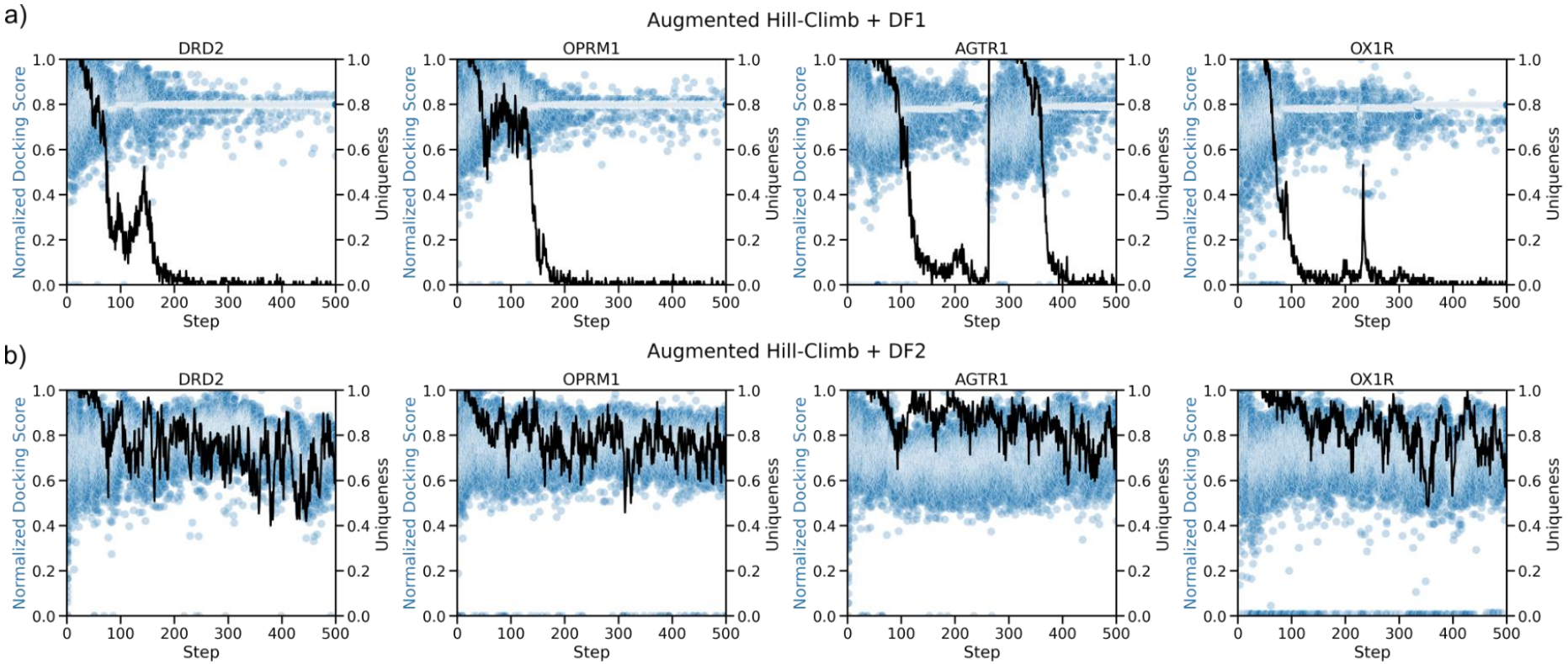


Figure 4.8: Optimization of normalized docking score and uniqueness during optimization across targets. (a) With diversity filter 1 (DF1), docking score converges to the minimum score threshold (0.8) of the DF and model undergoes mode collapse seen by an associated drop in uniqueness. (b) With diversity filter 2 (DF2), no convergence is observed, and uniqueness maintains relatively high. This is due to a lower minimum score threshold (0.5) and softer penalization scheme.

Table 4.5: Number of optimization steps taken before the mean docking score exceeds different internal and external thresholds (earliest sample exceeding threshold is shown in brackets). The final row lists the Augmented Hill-Climb in combination with DF2 fold improvement over REINVENT. Where a threshold was not reached within the maximum number of training steps (or samples) it has been annotated as being greater than 500 (or 32,000).

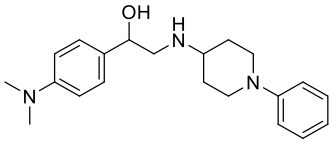
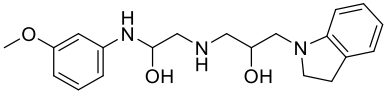
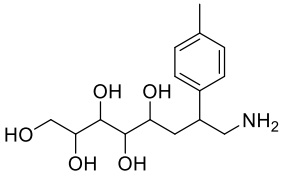
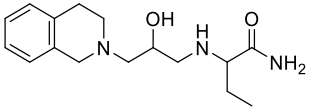
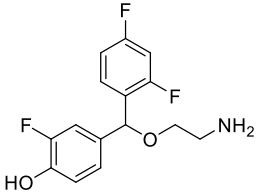
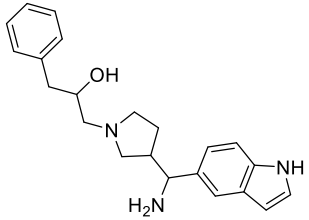
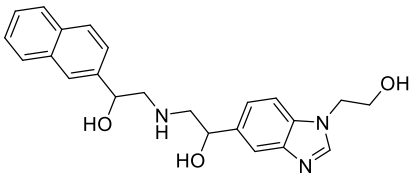
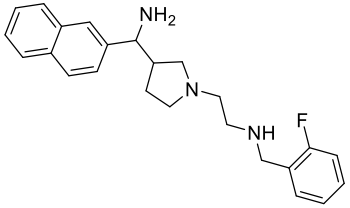
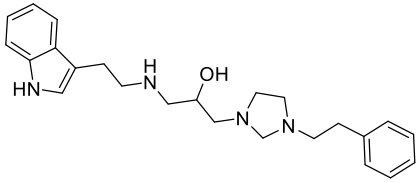
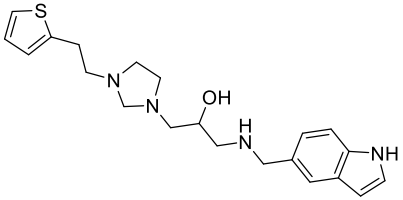
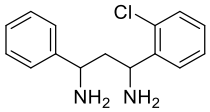
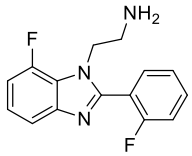
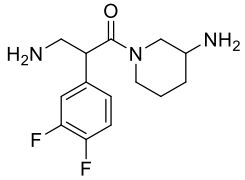
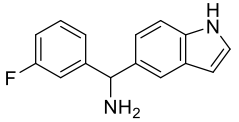
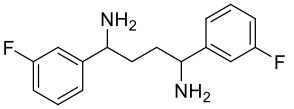
		Number of steps required for optimization beyond prior at a given threshold					Number of steps required for optimization beyond external thresholds		
	Threshold	120%	140%	160%	180%	200%	Inactive mean	Active mean	80% precision threshold
DRD2	REINVENT	> 500 (15)	> 500 (685)	> 500 (22,292)	> 500 (> 32,000)	> 500 (> 32,000)	1 (1)	163 (15)	> 500 (15)
	AHC+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)
	AHC+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)
	AHC+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)
	AHC+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)

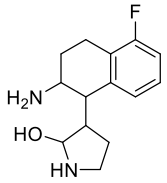
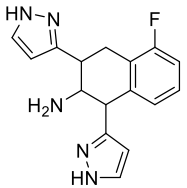
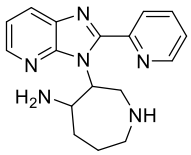
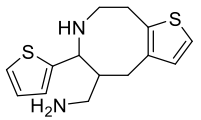
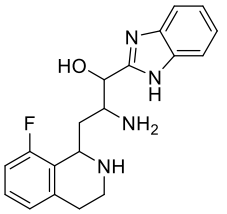
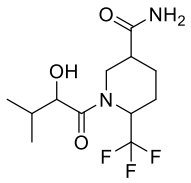
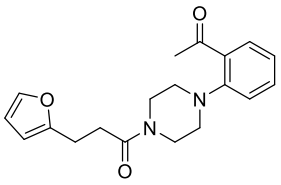
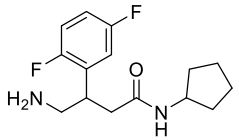
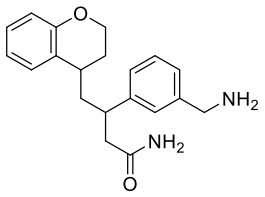
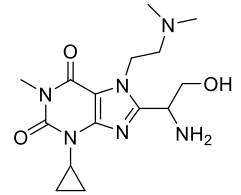
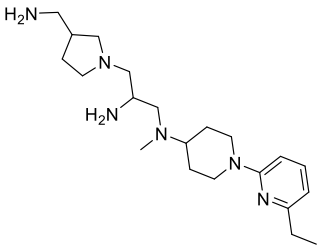
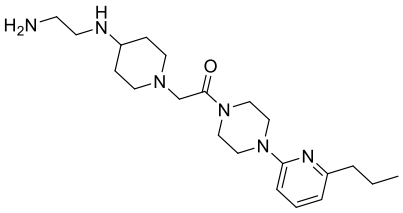
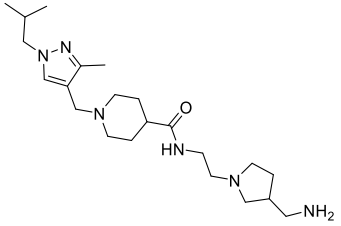
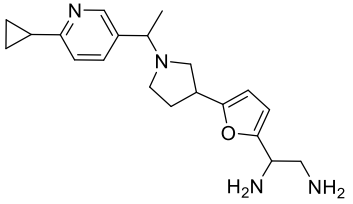
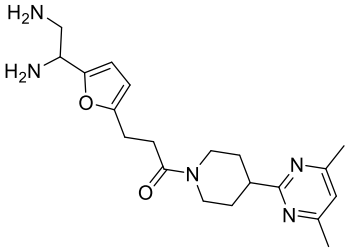
To investigate if similar chemistry was generated by the RL strategies, the top 500 scaffolds generated by REINVENT for each target are plotted, and at what stage they were first generated by either RL strategy, shown in Figure 4.7 (bottom panel of each sub-figure). This shows a general trend where AHC in combination with DF2 tends to generate scaffolds appearing in REINVENT at a later stage much sooner, and scaffolds appearing early in REINVENT much later. 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.

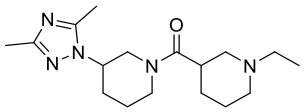
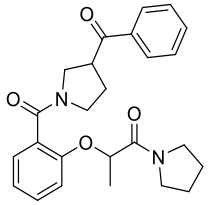
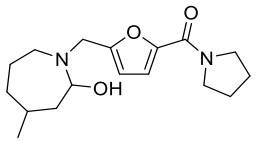
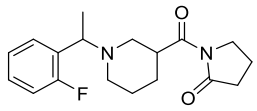
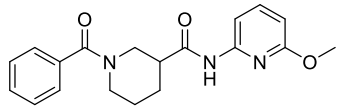
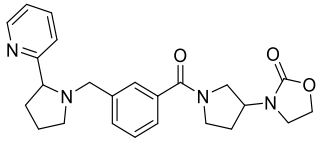
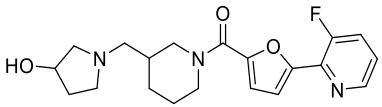
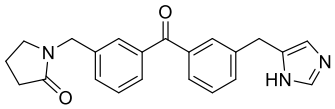
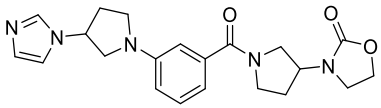
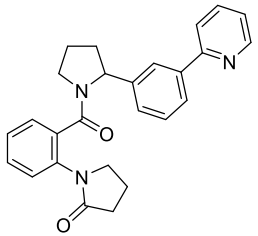
A visual comparison of the centroids of the top 100 compounds for each target for AHC in combination with DF2 and REINVENT is shown in Table 4.6. With disregard to prior knowledge of target ligands and suitability of the scoring function, the quality of chemistry generated is again indistinguishable 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 no acid moieties are generated for AGTR1 by either strategy which will be in part due to the docking algorithm targeting only the Y35^{1x39} sub-pocket and out towards the extracellular surface (Figure 4.9) as opposed to the sub-pocket surrounding R167^{4x65} as required for ligand activity[300].

In addition, Figure 4.10 shows that property space occupied by AHC *de novo* molecules is still maintained (mean remains within training set space) 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[301]. 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 docking 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.

Table 4.6: 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).

Target	RL strategy	Top 1	Top 2	Top 3	Top 4	Top 5
DRD2	REINVENT	 CS: 6 DS: -9.89 AvDS: -10.03	 CS: 4 DS: -9.89 AvDS: -10.22	 CS: 3 DS: -9.84 AvDS: -10.0	 CS: 3 DS: -9.84 AvDS: -9.88	 CS: 3 DS: -9.85 AvDS: -9.93
DRD2	AHC+DF2	 CS: 8 DS: -12.19 AvDS: -12.31	 CS: 3 DS: -11.69 AvDS: -12.05	 CS: 3 DS: -11.81 AvDS: -12.09	 CS: 3 DS: -11.83 AvDS: -11.95	 CS: 3 DS: -11.93 AvDS: -12.12
OPRM1	REINVENT	 CS: 16 DS: -8.72 AvDS: -8.90	 CS: 8 DS: -8.79 AvDS: -9.0	 CS: 3 DS: -8.72 AvDS: -8.82	 CS: 3 DS: -8.74 AvDS: -8.79	 CS: 3 DS: -8.77 AvDS: -8.80

OPRM1	AHC+DF2					
		CS: 3 DS: -9.47 AvDS: -9.54	CS: 3 DS: -9.53 AvDS: -9.65	CS: 2 DS: -9.49 AvDS: -9.59	CS: 2 DS: -9.49 AvDS: -9.58	CS: 2 DS: -9.49 AvDS: -9.54
AGTR1	REINVENT					
		CS: 2 DS: -8.73 AvDS: -8.73	CS: 2 DS: -8.76 AvDS: -8.87	CS: 2 DS: -8.76 AvDS: -8.85	CS: 2 DS: -8.82 AvDS: -8.90	CS: 2 DS: -8.90 AvDS: -8.94
AGTR1	AHC+DF2					
		CS: 4 DS: -11.72 AvDS: -12.11	CS: 4 DS: -12.48 AvDS: -12.55	CS: 3 DS: -11.87 AvDS: -11.95	CS: 3 DS: -11.88 AvDS: -12.01	CS: 2 DS: -11.72 AvDS: -11.82

OX1R	REINVENT					
		CS: 2 DS: -8.52 AvDS: -8.52	CS: 2 DS: -8.57 AvDS: -9.04	CS: 2 DS: -8.56 AvDS: -8.57	CS: 2 DS: -8.60 AvDS: -8.73	CS: 2 DS: -8.79 AvDS: -8.80
OX1R	AHC+DF2					
		CS: 10 DS: -10.01 AvDS: -10.24	CS: 9 DS: -9.85 AvDS: -9.98	CS: 5 DS: -9.98 AvDS: -10.05	CS: 4 DS: -9.92 AvDS: -10.0	CS: 3 DS: -9.87 AvDS: -10.0

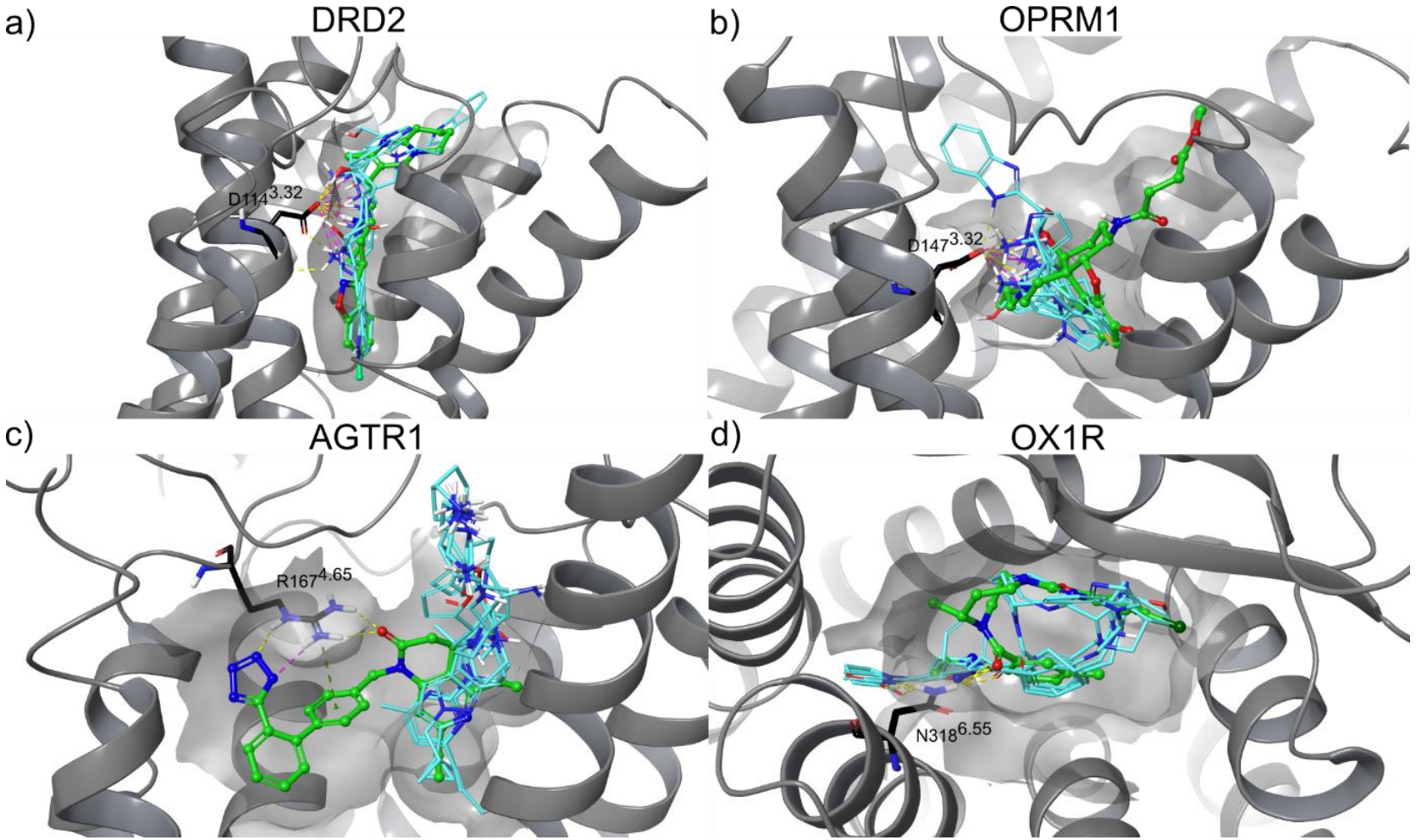


Figure 4.9: Docked poses of the centroid molecules (cyan) shown in Table 4.6 generated by AHC + DF2 compared to the co-crystallized ligand (green) for each respective target. For (a) DRD2 and (b) OPRM1, RL and docking results in sensible pose generation satisfying crucial residue interactions with D114^{3x32}. While for (c) AGTR1 poses occupy only one relevant sub-pocket compared to the co-crystal ligand and form no interactions with R167^{4x65}. Only one pose for (d) OX1R mimics the horseshoe shape adopted by the co-crystal ligand, however, most form an interaction with N318^{6x55}.

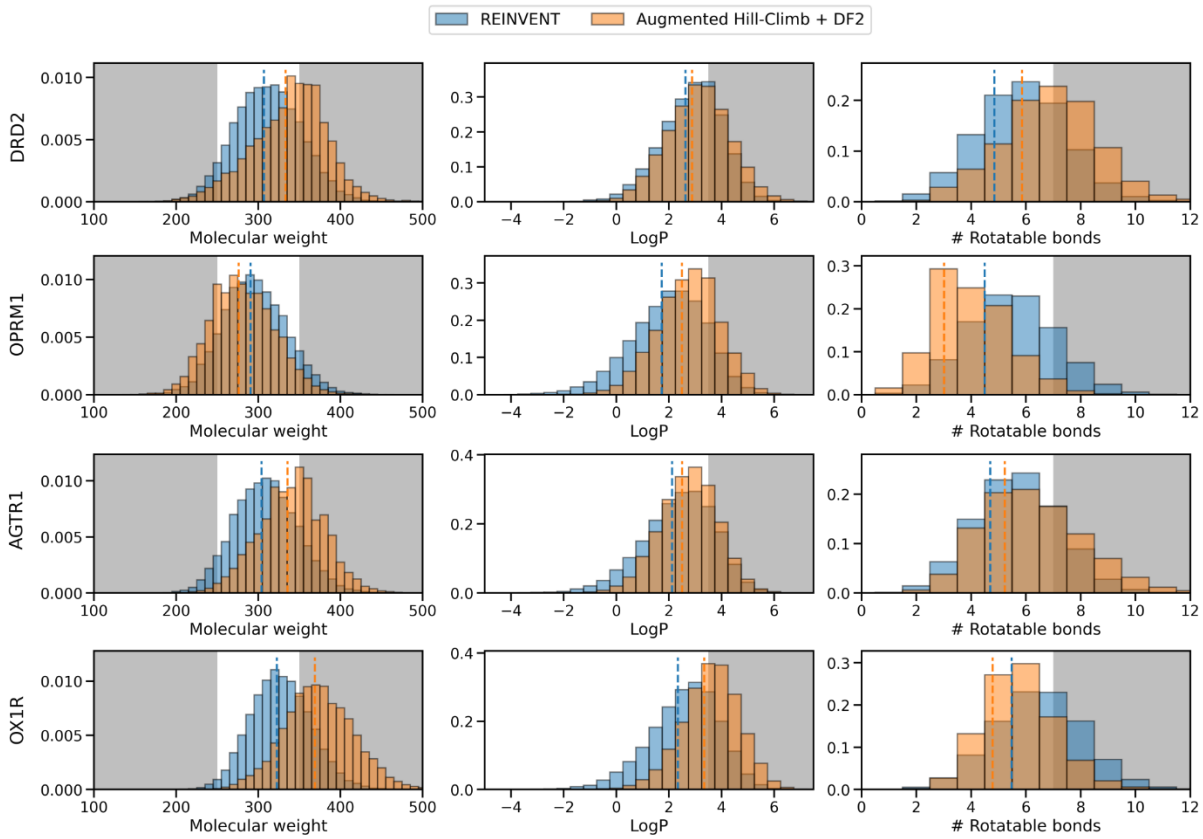


Figure 4.10: Property space comparison between REINVENT compared to Augmented Hill-Climb (in combination with DF2). Property space is according to molecular weight, LogP and the number of rotatable bonds for molecules optimized to minimize the docking score against the targets. The grey shading indicates property space not represented in the prior training set.

4.3.3 Effect of Augmented Hill-Climb diversity filter hyperparameters on molecule generation

Given the drop in uniqueness observed in Figure 4.8a, a hyperparameter search to identify optimal diversity filters and respective hyperparameters that best combat mode collapse was conducted (Experiment 3, Figure 4.1). DF configurations were tested on three representative objectives taken from the GuacaMol benchmark suite[54] with an RNN architecture, and training regime identical to that implemented in GuacaMol[296], with the exception of using AHC for optimization. This resulted in 825 individual runs across the three objective tasks that were assessed by computing the area under the curve during optimization for uniqueness, score/goal achieved as well as, taking the final run time.

In all cases (Figure C.1 - Figure C.3), a higher minimum score threshold (> 0.5) lead to poorer performance. The higher the minimum score threshold, the fewer molecules the DF is applied to and therefore the closer AHC is to being run without a DF,

explaining the drop in uniqueness as observed previously. The specific implementation of scoring functions used has a duplicate lookup function that may result in longer run times if many duplicate molecules are observed, explaining the counter-intuitive increase in run time with less actual DF use (higher minimum score threshold).

With respect to improving uniqueness – the main symptom of mode collapse - lower bin sizes, linear output mode and compound similarity/scaffold similarity (atom pair) DFs appear to perform best. Lower bin size corresponds to quicker penalization for certain chemotypes, although bin size effect is lesser for the Osimertinib MPO task. In the case of Osimertinib MPO (Figure C.3), simply penalizing non-unique molecules provides reasonable performance improvement from 0.19 AUC (no DF) to 0.87 AUC. Meanwhile linear output performs best when bin size is greater than 0 (note when bin size is 0 all output modes are effectively binary), suggesting that greater performance is achieved with a more gradual penalization gradient. Lastly, compound similarity and scaffold similarity (atom pair) DFs slightly outperform all others. These DFs are a softer measure of similarity than identical scaffolds or scaffold similarity (ECFP) (which has a higher minimum similarity threshold) resulting in more molecules being identified as similar and therefore penalized. Note that the minimum similarity threshold or fingerprint hyperparameters are left as default[167]. Preventing mode collapse and improving uniqueness typically requires stricter diversity filter parameters that penalize duplicated or similar molecules more easily, while a softer gradient of penalization is preferred.

With respect to the objective score, there was less discrepancy between output modes and the bin size and observations effectively reversed. Higher bin sizes and the stricter measures of similarity (identical murcko scaffold and scaffold similarity (ECFP)) showed higher AUCs indicating better performance. These more lenient diversity filter hyperparameters likely enable AHC more time to associate chemotypes with high rewards resulting in increased objective scores.

Overall, a trade-off is required in choosing DFs and hyperparameters for use in combination with AHC. DF penalization must be strict enough to reliably prevent mode collapse as observed by a drop in uniqueness, yet lenient enough to enable AHC to learn chemotype-reward associations. These observations led to the design of DF2

which is a compromise between preventing mode collapse and achieving high objective scores.

4.3.4 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[77], REINVENT[76,99], BAR[123] and Hill-Climb[54], as well as in combination with KL regularization for non-regularized strategies (Experiment 4, Figure 4.1). In the interest of standardisation, the prior was trained on the GuacaMol train dataset. The RL strategies were applied to six tasks of varying practical difficulty (see 4.2.7.4). DF2 was used in all cases except for the Risperidone similarity task which uses a lower minimum score threshold of 0 due to low similarity values observed (DF3, Table 4.2).

The performance of task optimization is shown in Figure 4.11. AHC is the most efficient of all RL strategies at all tasks except for maximizing the number of heavy atoms. It is particularly better than the other RL strategies during early-stage optimization (e.g., Figure 4.11) and in more difficult objectives (e.g., Figure 4.11e,f). AHC even outperforms un-regularized RL strategies. 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, the same smaller batch size with HC was investigated, 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 (Figure C.4 and Figure C.5). 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, re-implementation of BAR performed particularly poorly in most cases except for DRD2 activity (the case study in the original implementation[309]). 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).

The efficiency benefit of AHC is true also by wall time (Figure C.6). To put this practical benefit into greater context, Table 4.7 shows the CPU hours required to reach different optimization thresholds for the DRD2 docking score task. AHC is the only strategy able to optimize the mean docking score to 180% and 200% that of the initial prior mean docking score within the given time. Moreover, AHC also achieves lower optimization thresholds much quicker, for example, 140% in just 16 hours compared to 202 hours for REINVENT 2.0. This optimization task was parallelized over 10 CPUs and therefore actually corresponded to 1.6 hours and 20.2 hours respectively. Given access to just 10 CPUs, AHC is able to achieve 200% optimization from the prior in less than a day (21.6 hours). This enables optimization tasks to be run on single, local machines (e.g., 6-12 CPUs) on a far more reasonable time scale than previously possible, without the need for cloud computing. This provides opportunities for more than one expensive scoring function (e.g., docking into two receptors, or docking and computer-aided synthesis planning) to be used to evaluate molecule fitness on a more reasonable time scale.

Figure C.7 to Figure C.12 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[54] training set e.g., increasing molecular weight seen in the DRD2 docking score optimization task (Figure C.10). This is even observed for the more heavily regularized REINVENT strategy but is not present when using the MOSES[n] training set (Table 4.6). Moreover, KL regularization as proposed previously[122,125] does not seem to improve chemistry generated by REINFORCE and HC and instead shows a tendency to increase molecular weight (Figure C.9). 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 C.11), is less prone to idiosyncratic tendencies of HC (e.g., large molecules and long chains in Figure C.11), yet more sample efficient than either. Overall, 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*.

1

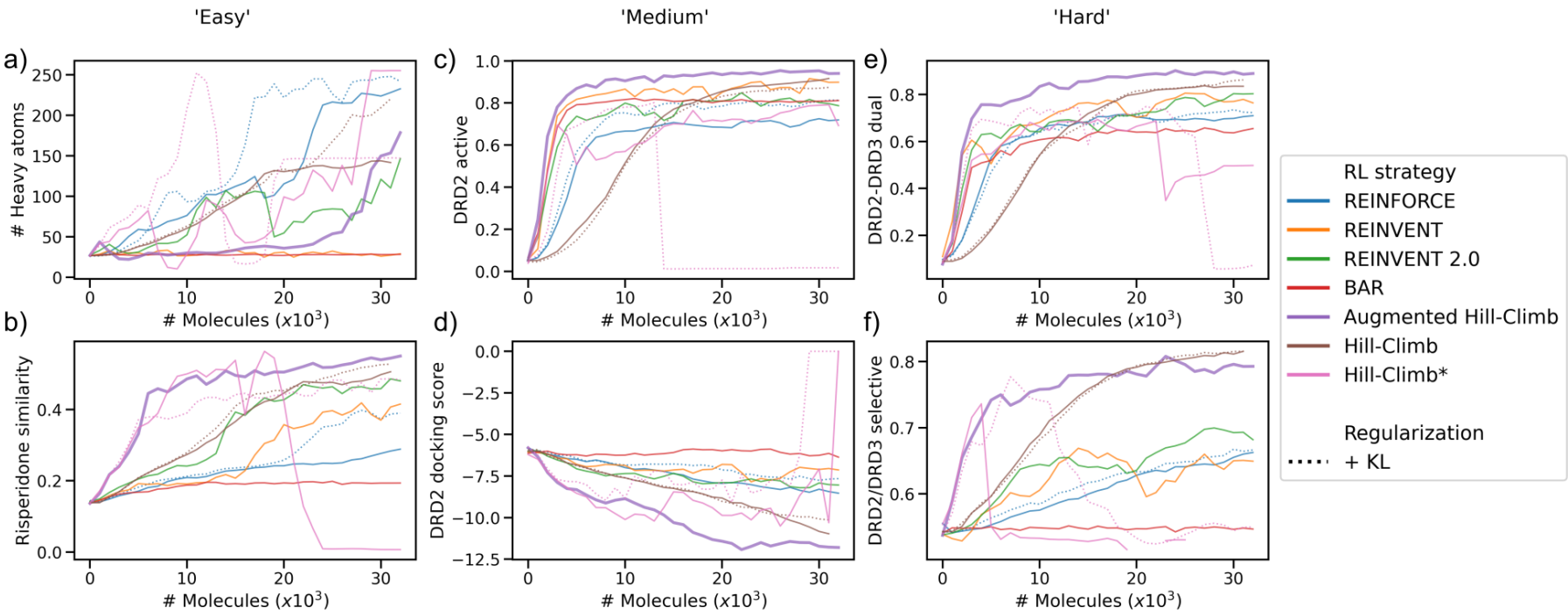


Figure 4.11: Per-molecule optimization of different RL strategies against different objective tasks of varying difficulty: (a) number of heavy atoms, (b) Similarity to Risperidone (DRD2 inverse agonist), (c) predicted probability of DRD2 activity, (d) Glide-SP docking score against DRD2, (e) predicted 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.

Table 4.7: CPU hours required for RL strategies to optimize the DRD2 docking score benchmark task to different thresholds. Time is representative of when the batch mean exceeds the respective internal / external threshold (time of the earliest sample exceeding threshold is shown in brackets). Run using an AMD Threadripper 1920x CPU and Nvidia GeForce RTX 2060 super GPU. Failing to reach a threshold is marked by a "-".

	CPU hours required for optimization beyond prior at a given threshold					CPU hours required for optimization beyond external thresholds		
Threshold	120%	140%	160%	180%	200%	Inactive mean	Active mean	80% precision
REINFORCE	74 (0)	173 (0)	- (20)	- (34)	- (96)	2 (0)	103 (0)	177 (0)
REINFORCE + KL regularization	183 (0)	- (0)	- (33)	- (74)	- (216)	22 (0)	204 (0)	- (0)
REINVENT	79 (0)	- (0)	- (8)	- (164)	- (-)	4 (0)	93 (0)	- (0)
REINVENT 2.0	38 (0)	202 (0)	- (16)	- (53)	- (92)	12 (0)	51 (0)	198 (0)
BAR	- (0)	- (0)	- (32)	- (32)	- (-)	4 (0)	0 (0)	- (0)
Hill-Climb	44 (0)	114 (0)	177 (0)	218 (24)	- (85)	16 (0)	57 (0)	99 (0)
Hill-Climb + KL regularization	45 (0)	106 (0)	157 (0)	- (45)	- (45)	8 (0)	58 (0)	99 (0)
Hill-Climb*	11 (0)	31 (1)	52 (6)	- (15)	- (31)	2 (0)	11 (0)	24 (0)
Hill-Climb* + KL regularization	14 (0)	28 (0)	74 (1)	- (17)	- (17)	6 (0)	17 (0)	31 (0)
Augmented Hill-Climb	9 (0)	16 (0)	72 (0)	151 (14)	216 (15)	2 (0)	13 (0)	27 (0)

4.3.5 Applying Augmented Hill-Climb to transformer architectures

RL algorithms (including AHC) should be model-agnostic and therefore applicable to other models used in a policy-based reinforcement learning setting. To test this and confirm whether AHC is still more sample efficient than the baseline REINVENT in this setting, both strategies were applied to a Transformer model (Tr) that uses state-of-the-art attention mechanisms[84] for AR sequence generation (Experiment 5, Figure 4.1). To better understand any underlying difference in model behaviour, performance metrics were calculated, shown in Table 4.8 and Table 4.9. Although the Tr model didn't undergo extensive hyperparameter optimization, performance differences are marginal for both intrinsic properties and extrinsic properties in comparison to an RNN. Therefore, the underlying ability to learn a chemical distribution is comparable and suitable for further comparison of RL behaviour.

RL was then conducted using the same approach as with the RNN on the same DRD2-based benchmark applied previously, for both REINVENT and AHC in combination with DF3. Figure 4.12 shows that AHC still outperforms REINVENT with regards to sample efficiency and optimization power. However, as shown in Figure 4.12a-c,e the Tr model is much less stable under RL optimization compared to the RNN and more readily undergoes mode collapse i.e., it starts generating invalid or repeated molecules, as shown in Figure C.13 and Figure C.14. In fact, very few implementations of transformers exist within a RL setting[97,310]. Therefore a modified transformer architecture designed to stabilize model optimization during RL[297] was also implemented. This gated transformer (GTr) architecture implements a GRU-like gate in-place of the residual connection and relocates layer normalization to input streams (notably this is not the only recent example of combining concepts from GRUs or LSTMs with transformer architectures[311]). As shown in Figure 4.12, this appeared to stabilize RL and again showed that AHC outperforms REINVENT with respect to sample efficiency, leaving only the heavy atom task still failing with AHC which is notably outside the applicability domain of the training dataset (and also devoid of any practical relevance). Examples of *de novo* chemistry generated by these models can be seen in Figure C.15 to Figure C.20. Overall, this shows that RL sample efficiency gains by AHC also generalize to other language models.

Table 4.8: Intrinsic properties of 10,000 sampled *de novo* molecules from the recurrent neural network (RNN), Transformer (Tr) and Gated Transformer (GTr) when trained on the GuacaMol training dataset.

Model	**Valid** (↑)	**Unique** (↑)	**Novel** (↑)	**SEDiv@1k** (↑)	**Scaffold Unique** (↑)	**FG** (↑)	**RS** (↑)	**MOSES Filters** (↑)
RNN	0.96	0.99	0.96	0.88	0.84	0.20	0.18	0.52
Tr	0.96	1.00	0.97	0.85	0.86	0.18	0.17	0.52
GTr	0.94	0.99	0.97	0.89	0.87	0.21	0.18	0.50

Table 4.9: Extrinsic properties of 10,000 sampled *de novo* molecules from the recurrent neural network (RNN), Transformer (Tr) and Gated Transformer (GTr) when trained on the GuacaMol training dataset. These metrics measure the similarity of *de novo* molecules to the GuacaMol test dataset.

Model	**Analogue Similarity** (↑)	**Analogue Coverage** (↑)	**SNN** (↑)	**FG** (↑)	**RS** (↑)	**Frag** (↑)	**Scaff** (↑)	**logP** (↓)	**NP** (↓)	**SA** (↓)	**QED** (↓)	**Weight** (↓)
RNN	0.83	0.43	0.52	1.00	1.00	1.00	0.64	0.02	0.04	0.08	0.01	13.01
Tr	0.82	0.40	0.51	1.00	1.00	1.00	0.59	0.07	0.08	0.04	0.01	2.90
GTr	0.76	0.36	0.49	1.00	1.00	1.00	0.55	0.07	0.07	0.03	0.00	3.47

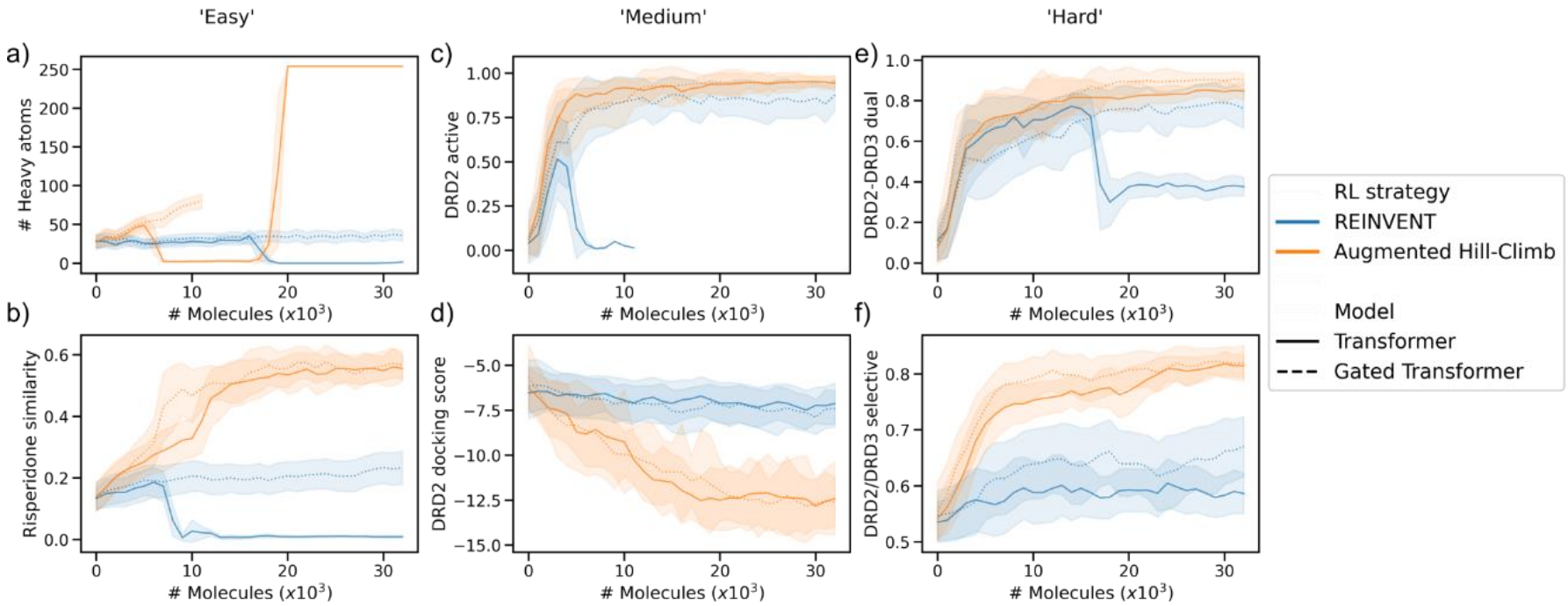


Figure 4.12: Per-molecule optimization by REINVENT and Augmented Hill-Climb RL strategies for the transformer (Tr) and gated transformer (GTr) architecture against the DRD2 benchmark objectives. Tr is more unstable during RL by REINVENT which is stabilized by the GTr. In all cases Augmented Hill-Climb outperforms REINVENT at objective optimization. Although these transformer models are more prone to mode collapse than an RNN as observed by a drop in validity and uniqueness as shown in Figure C.13 and Figure C.14.

4.3.6 Benchmarking Augmented Hill-Climb against other generative models

Gao et al.[55] recently proposed a practical molecular optimization (PMO) benchmark focussing on generative model sample efficiency. In this, they compared a wide range of 22 generative models (run with five replicates) on a selection of 23 commonly used objective tasks (QED[173]; predicted DRD2[76], GSK3β[312] and JNK3[312] activity; and 19 from GuacaMol suite[54]). The authors apply a limit of maximizing the objective within a fixed budget (10,000 oracle evaluations) and measure performance by comparing the area under the curve (AUC) of the average top 10 molecules during optimization. They found that REINVENT[76] was the most sample efficient across tasks, albeit using a σ value of 500, much higher than explored here. AHC has already demonstrated increased sample efficiency compared to REINVENT, suggesting potential state-of-the-art performance relative to other generative models as well. However, this needed to be investigated further, especially with regards to the value of σ used and its consequences.

Based on the observations in 4.3.1, the use of high σ values for REINVENT are particularly concerning regarding the extrapolation of chemistry outside of the training dataset chemical space. To investigate, the properties and structures of top ranking *de novo* molecules for a particularly well performing objective (JNK3β) are shown in Figure 4.13a. This shows that in 4/5 replicate runs, the top 10 molecules have molecular weight and LogP distributions far beyond the training dataset, and typically contained 0-10% ECFP4 bits unobserved in the training dataset. Moreover, visualizing the top 2 *de novo* molecules for each run highlights undesirability from a chemical perspective, with large molecules and many repeating substructures. Therefore, it is clear that in its current form the benchmark is limited to sample efficiency only, and the results should be interpreted with caution as they do not adequately account for any practical consideration of chemistry. Overall, it must reflect other desirable properties of de novo molecule generation to serve as a meaningful comparison between generative models.

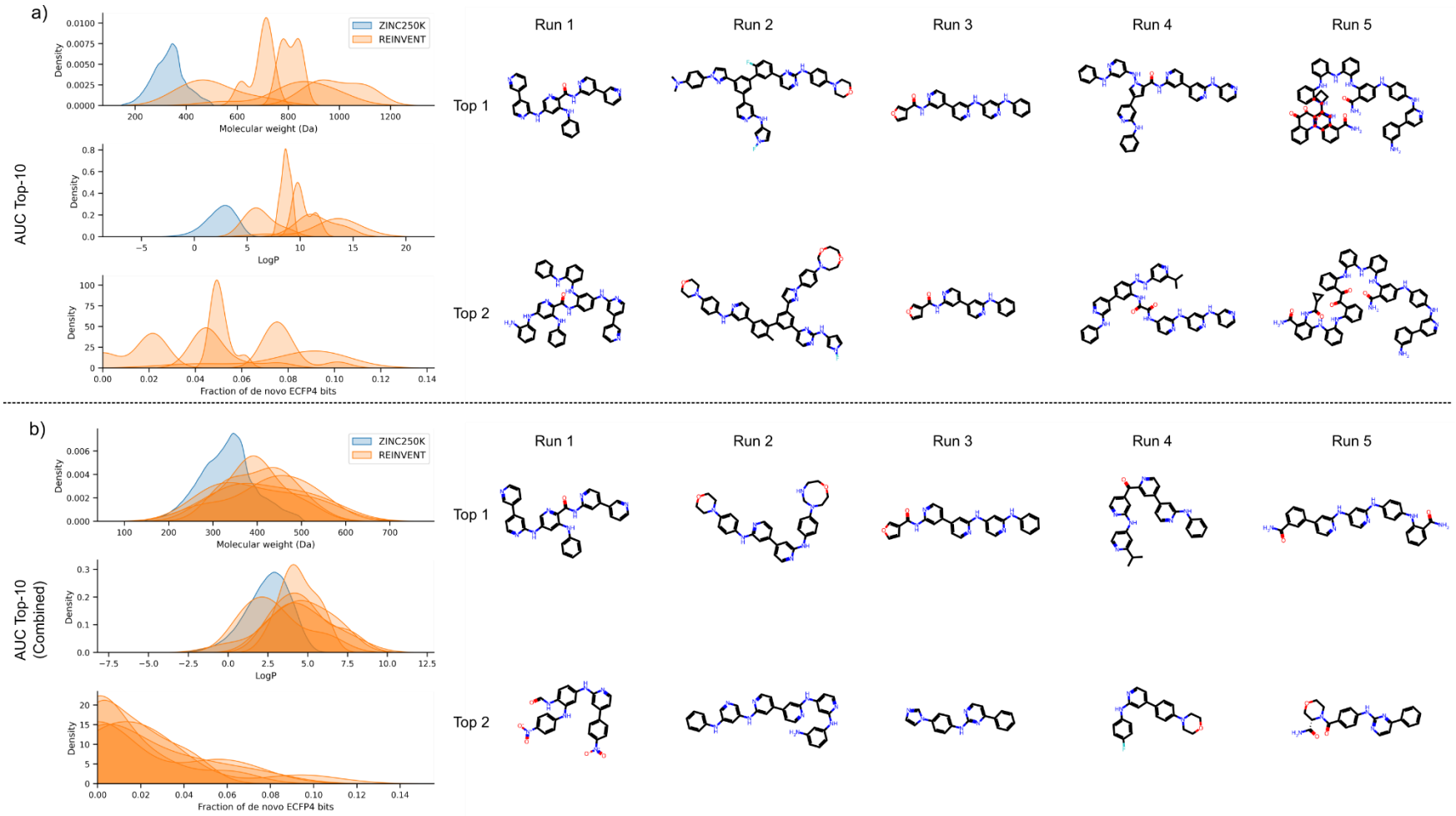


Figure 4.13: Property distribution and example REINVENT *de novo* molecules from the JNK3β task in PMO benchmark. (a) Top molecules when by score alone. Left: Molecular weight, LogP and fraction of outlier bits of the top 10 *de novo* molecules relative to the ZINC250k training dataset for each replicate. Right: Top 2 molecules are displayed from each replicate run. (b) The same is shown but when applying property filter and diversity filters to the top 10 molecules.

To better account for chemistry of *de novo* molecules, three new endpoint metrics to were implemented to re-evaluate generative model performance in this benchmark (see 4.2.8). The first metric, AUC Top-10 (Filtered), ensures the molecular weight and logP of the top ten molecules are within four standard deviations of the training dataset and contain less than 10% outlier bits. The second metric, AUC Top-10 (Diverse), ensures the top ten molecules are diverse with respect to each other. The final metric, AUC Top-10 (Combined), applies both of these constraints. These metrics are purposefully lenient due to subjectivity in defining sensible chemistry (which also depends on the training dataset used as reference), as well as the evaluation bias in favour distribution-based models, as rule-based models have no explicit chemical space constraints. However, these constraints could be made stricter, which is explored later.

The extent to which *de novo* molecules violate these constraints is shown in Figure 4.14. This shows that for most methods, <50% of *de novo* molecules pass the lenient molecular weight, logP and outlier bit constraint proposed here. The same is observed with respect to diversity, where most methods contain <50% molecules considered diverse as defined here, including REINVENT. It should be noted that very high diversity values are difficult to interpret alone, as *de novo* molecules with idiosyncratic structures artificially inflate diversity measures. Interpreting together with outlier ECFP4 bits will capture such a scenario or comparing in reference to the training dataset as a baseline (‘screening’). In this case, ~50% of molecules randomly screened from ZINC250k are considered diverse, and so values in this vicinity or slightly higher are desirable. Applying the combination of these constraints to the endpoint metric i.e., top ten molecules, restricts selection to be more chemically reasonable, as shown in Figure 4.13b.

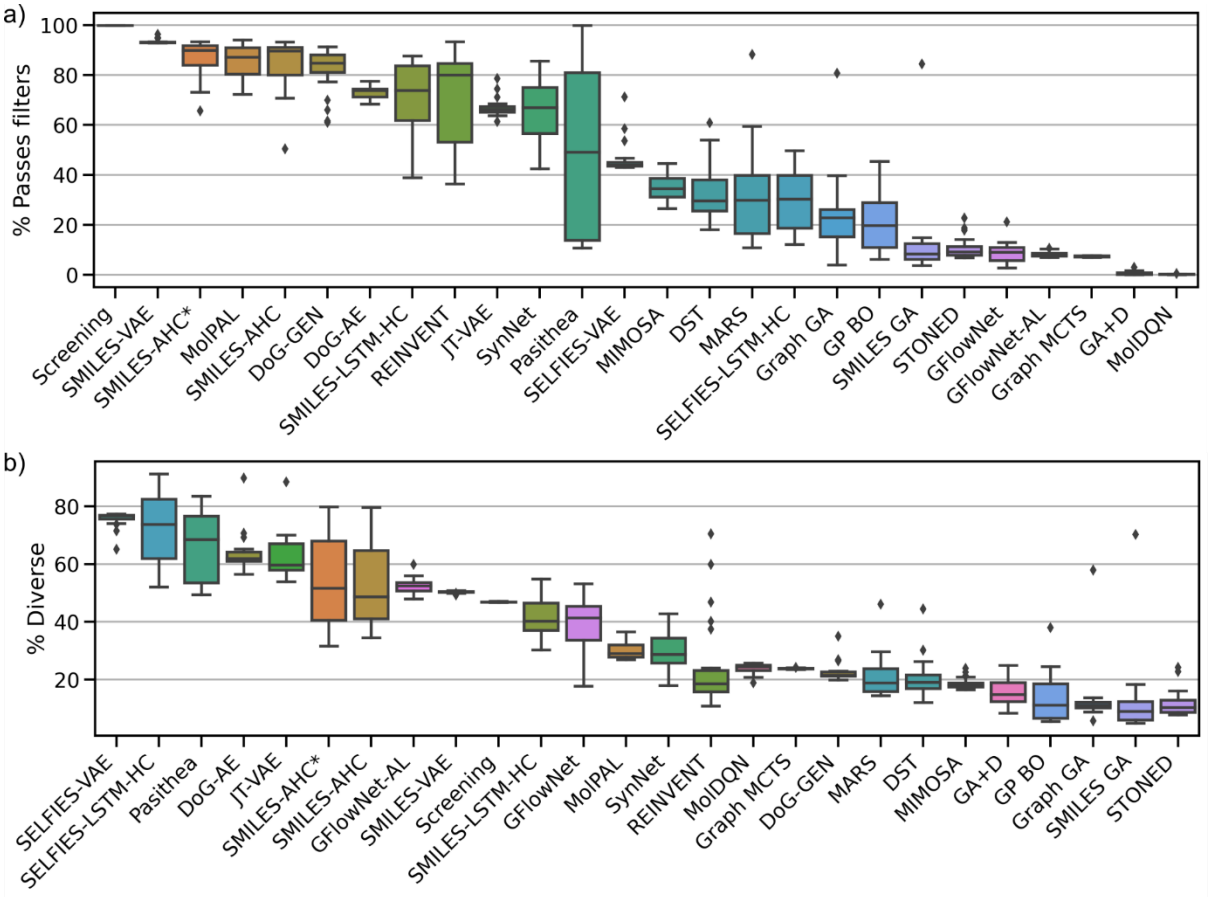


Figure 4.14: Percent of *de novo* molecules generated in the PMO benchmark that either (a) pass property constraint and outlier ECFP4 bit constraint or (b) are sufficiently diverse. Note that Screening and MolPAL are techniques that sample directly from the training dataset and do not generate *de novo* molecules.

To test AHC, following the protocol of the PMO benchmark, an RNN was trained on the proposed training data (ZINC250k) and AHC hyperparameter optimization was run on the same two objective tasks used in PMO. Although Figure 4.15 shows the hyperparameters with the best overall average AUC on the two tasks is a batch size of 256, σ of 120 and *k* of 0.25, other combinations are almost as performant indicating some robustness to hyperparameter choice. This combination of hyperparameters was selected for benchmarking (SMILES-AHC), as well as the default σ value used in previous sections with which there is already some understanding of behaviour (SMILES-AHC*). These results are already in contrast to REINVENT where the highest tested value 500 for σ far outperformed any other hyperparameter combination[55]. Note no diversity filter is used with AHC to provide a standardized comparison despite it being shown to improve performance in 4.3.3.

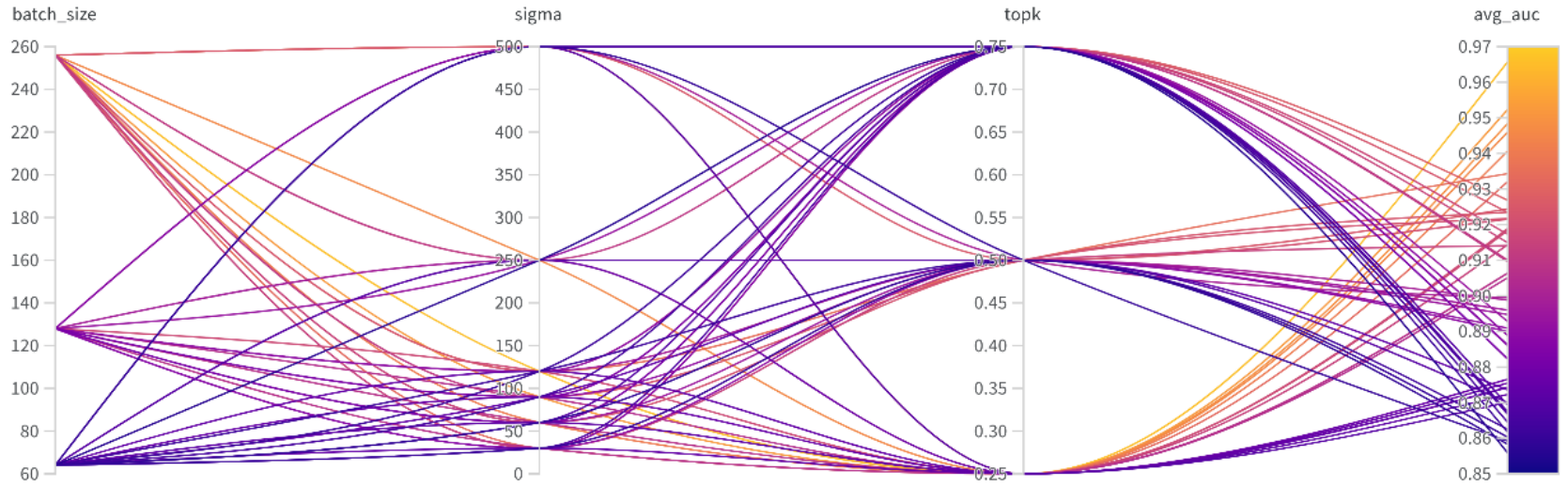


Figure 4.15: Hyperparameter optimization of Augmented Hill-Climb hyperparameters on the two test objectives Zaleplon MPO and Perindopril MPO in the PMO benchmark.

AHC generated *de novo* molecules are within the top two generative models that adhere to training dataset constraints (see Figure 4.14a) and within the top 10 generative models with respect to diversity (see Figure 4.14), albeit some of these may be artificially inflated as mentioned previously. However, in both cases AHC outperforms REINVENT, likely due to the incredibly high σ value used with REINVENT. Revisiting the original benchmark results by AUC Top-10, REINVENT[76] is ranked 1st, followed by Graph-GA[56] and GP-BO[149], shown in Figure 4.16a. Integrating SMILES-AHC into the PMO benchmark results in a ranking of 5th. However, accounting for chemistry with respect to the training dataset improves the ranking to 2nd behind REINVENT, with Graph-GA dropping to 6th and GP-BO dropping to 4th. However, the best 10 *de novo* molecules that pass these chemistry filters span the first 290 compounds for AHC, compared to the first 2,723 compounds for REINVENT, on average. This means that many more *de novo* molecules are thrown out with REINVENT to identify the best 10, despite this REINVENT still ranks 1st. Upon making these chemical filters stricter to the point of requiring the same molecular weight, logP and bit presence as the training dataset, REINVENT performance declines relatively more than AHC, resulting in AHC ranking 1st with the strictest chemical constraints, as shown in Figure C.21. Meanwhile, accounting for diversity results in an AHC ranking of 2nd behind Graph-GA, with REINVENT dropping to 4th and GP-BO dropping to 8th. Finally, when accounting for both chemical constraints, AHC ranks 1st, REINVENT drops to 4th, Graph-GA to 7th and GP-BO to 9th. More drastic changes can be observed, for example, STONED[291] drops from 4th to 19th with the addition of chemical constraints proposed here. This highlights the potential for incredibly misleading interpretation of performance if chemistry is not considered. More generally, rule-based methods particularly suffer as there is no chemical distribution learning or explicit regularization.

Although AHC optimization and sample efficiency performance is strong on all tasks (see Figure 4.16), it appears to be markedly better at empirically more difficult objectives including isomer tasks, Zaleplon MPO and Sitagliptin MPO. These tasks may constitute low reward scenarios, which AHC empirically performs better in due to the hypothesized design.

Overall, when accounting for the type of chemistry generated *de novo* which is important for practical use cases, AHC achieves state-of-the-art performance compared to a variety of different generative model algorithms.

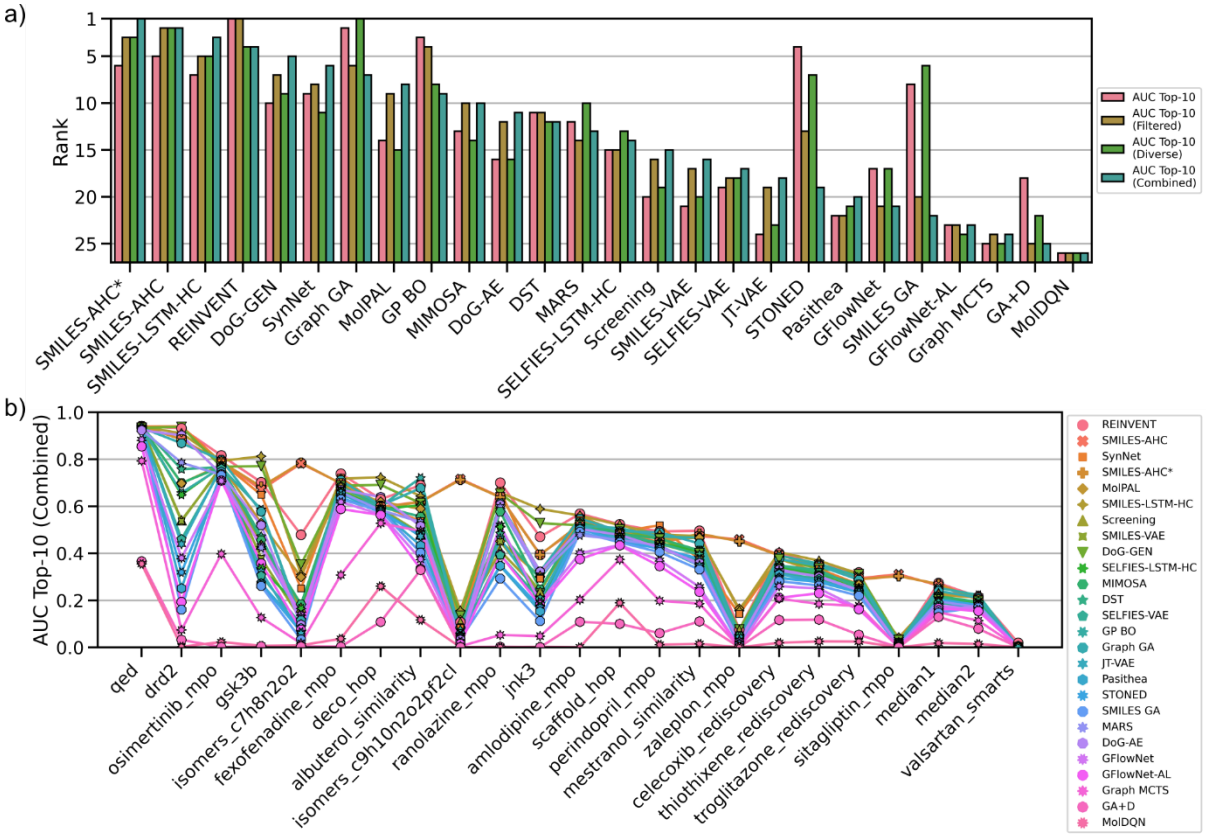


Figure 4.16: Sample efficiency performance of PMO benchmark with additional metrics introduced here. (a) Rank performance ordered by AUC Top-10 Combined. (b) AUC Top-10 Combined performance per benchmark objective.

4.4 Conclusion

In this chapter, I propose a modification to the REINVENT[76,99] RL framework for language-based *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[212] 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[54,77,122], 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. I hypothesize this is achieved by circumventing unwarranted regularization in REINVENT, but it can equally be viewed as applying essential regularization to the Hill-Climb algorithm. This algorithm can be successfully applied to transformer architectures showing that it generalizes across models. Furthermore, when compared to a broad variety of generative models, it achieves state-of-the-art sample efficiency when chemistry is accounted for in addition to sample efficiency. The improvement in sample efficiency enabled by Augmented Hill-Climb is especially useful when using computationally expensive scoring functions such as molecular docking or computer-aided synthesis planning tools.

Alternative methods can be used to improve the sample efficiency of RL[128]. For example, experience replay can be used to remind the agent of 'good' molecules[99,128], a margin guard[313] can be employed to dynamically change σ during RL updates or curriculum learning can be used to accelerate learning by breaking the objective into a sequence of simpler tasks[250]. However, AHC is a more principled approach to improve sample efficiency and could even be used in combination with these 'tricks' to potentially further improve RL for *de novo* molecule optimization.

Conclusions

The work presented in this thesis contributes an open-source Python software that is applied to evaluate: 1) integration of SBDD principles and 2) algorithmic changes to improve sample efficiency. Firstly, MolScore is a user friendly and configurable scoring function suite that includes, for example, interaction with five docking algorithms, pre-trained QSAR models on ~2,700 ChEMBL protein targets, functionality for transformation and aggregation of multiple values, a range of performance metrics, and a graphical user interface. Secondly, MolScore is used to implement and compare the difference in *de novo* molecules proposed by REINVENT when using ligand-based QSAR models or structure-based docking algorithms as proxy scoring functions. This identified that a structure-based approach can lead to more diverse recovery of known chemotypes and an increased ability to learn crucial residue interactions, such as with $D114^{3x32}$ in the D_2 receptor orthosteric pocket. Lastly, I propose AHC as a hybrid policy-based RL algorithm to further improve the sample efficiency (~45-fold) and optimization ability (~1.5-fold) when learning to optimize the docking score against four different GPCR receptors. Meanwhile ensuring the quality of chemistry proposed remains unchanged compared to the baseline REINVENT. This enables SBDD scoring functions to be optimized on a more practical timescale. Moreover, the results presented highlight that 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.

Looking forward, a lot of progress is needed to apply generative model algorithms to successfully accelerate drug discovery, in regard to all aspects including evaluation and benchmarking, methods, and objectives.

First of all, as exemplified in 4.3.6, many generative model benchmarks don't adequately consider the chemistry of *de novo* molecules. This is critical to be useful in practice, but also admittedly non-trivial to measure. *De novo* chemistry can either be measured objectively by applying known rules such as avoiding PAINS filters, or it can be measured based on similarity to a trusted, drug-like dataset. The latter is far easier, albeit very subjective to the composition of a drug-like dataset. Moreover, this results in unfair evaluation for rule-based models that don't learn this distribution or are not regularized by it. This could be resolved by adding physicochemical and similarity

properties as part of the objective[209] with the caveat that multiple parameters are more difficult to optimize than one. Either way, I firmly believe that chemistry should be considered as part of the evaluation or benchmark metric. Otherwise, we may advance the field based on misleading results in the wrong direction, resulting in a state of practical uselessness. The field also needs to continue its trajectory away from drug design irrelevant objectives like Penalized LogP, and towards those used in practice like binding affinity prediction, as in this thesis.

Aside from evaluating and benchmarking generative models against each other, the field sorely lacks a relative comparison to traditional approaches such as VS. Although generative models have the advantage of implicitly larger and quicker to search chemical spaces, VS libraries are growing rapidly and are usually make on demand thereby containing a degree of confidence in being able to test them in the real world. Moreover, techniques like ML-augmented VS[314,315] are rapidly increasing their search efficiency by only screening the most promising areas of chemical space, or combinatorial screening approaches[316,317] are being used based on available building blocks. Therefore, it would be of great value to identify the complementarity between these approaches and the situations in which it is better to apply one over the other.

Generative models have by now proven the ability to be practically useful and learn sensible distributions over 2D chemical space, as well as optimization towards certain endpoints. The next logical step is to move towards 3D molecule generation with respect to protein binding, in which *de novo* design was mostly originally conducted decades ago. There is now an increasing trend towards using ML to build molecules in 3D by learning conditional distributions of chemical structures relative to protein binding pockets with 3D-based generative models[249]. However, these models must learn from far more limited data and also appropriately handle equivariance, moreover, no 3D structure-based generative model has yet shown the ability to additionally optimize an objective, nor been prospectively validated.

As goal-directed generative models have proven the ability to maximize arbitrary numerical rewards, one of the keys limiting factors is now fast, accurate and robust scoring functions that avoid exploitation like shown by Renz et al.[134] or shown by large greasy molecules to exploit docking scoring functions in 4.3.4. Current state-of-the-art methods to evaluate molecule binding like molecular dynamics and free energy

perturbation are too computationally expensive to be used as scoring functions for goal-directed generative models and are still not robust enough to be used in an automated way. Over time these will become more accessible with increased compute power and algorithmic efficiency. Improvement in either predictive accuracy of binding affinity or efficiency of current methods will further unlock the potential of goal-directed generative models. Overall, this statement is equally applicable to any *in silico* model of drug properties such as toxicity, pharmacology, and pharmacokinetics. Subsequent improvements in these aspects will directly relate to improvement in *de novo* molecule generation by negating the need to avoid current scoring function limitations.

Appendix A

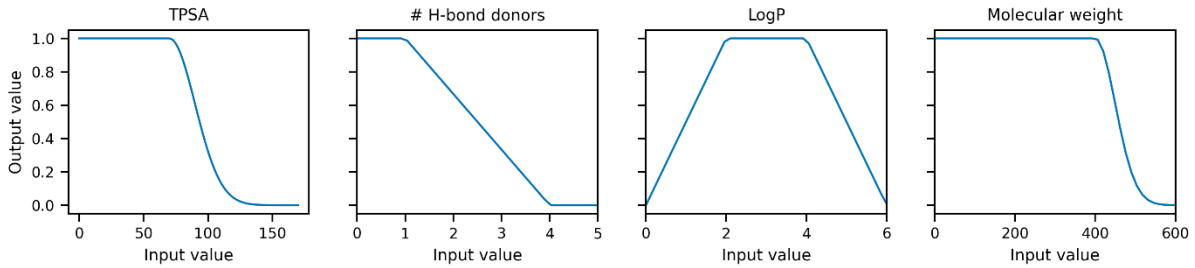


Figure A.1: Transformation functions mapping blood brain barrier molecular descriptors values into the range zero to one. From left to right: TPSA transformed by a Gaussian minimization function, # H-bond donors transformed by a linear threshold minimization function, LogP transformed by a linear threshold range function, and molecular weight transformed by a Gaussian minimization function.

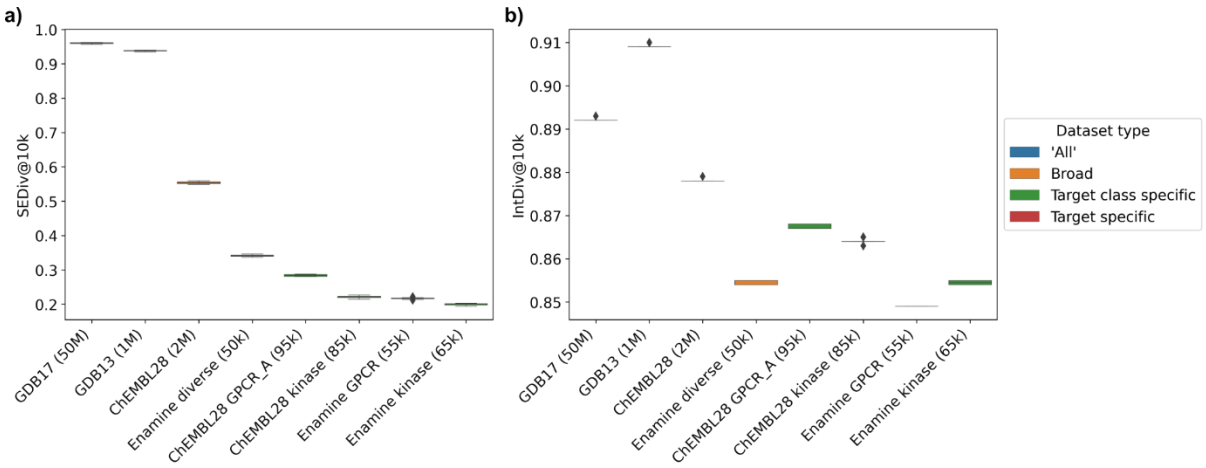


Figure A.2: The measured SEDiv (a) and IntDiv (b) of a randomly sampled 10,000 (@10k) subset of a variety of virtual libraries and datasets of characterised molecules with activity against particular targets belonging to a target class. IntDiv measures GDB13 as more diverse than GDB17, while SEDiv measures GDB17 as more diverse than GDB13 – in line with chemical intuition.

Appendix B

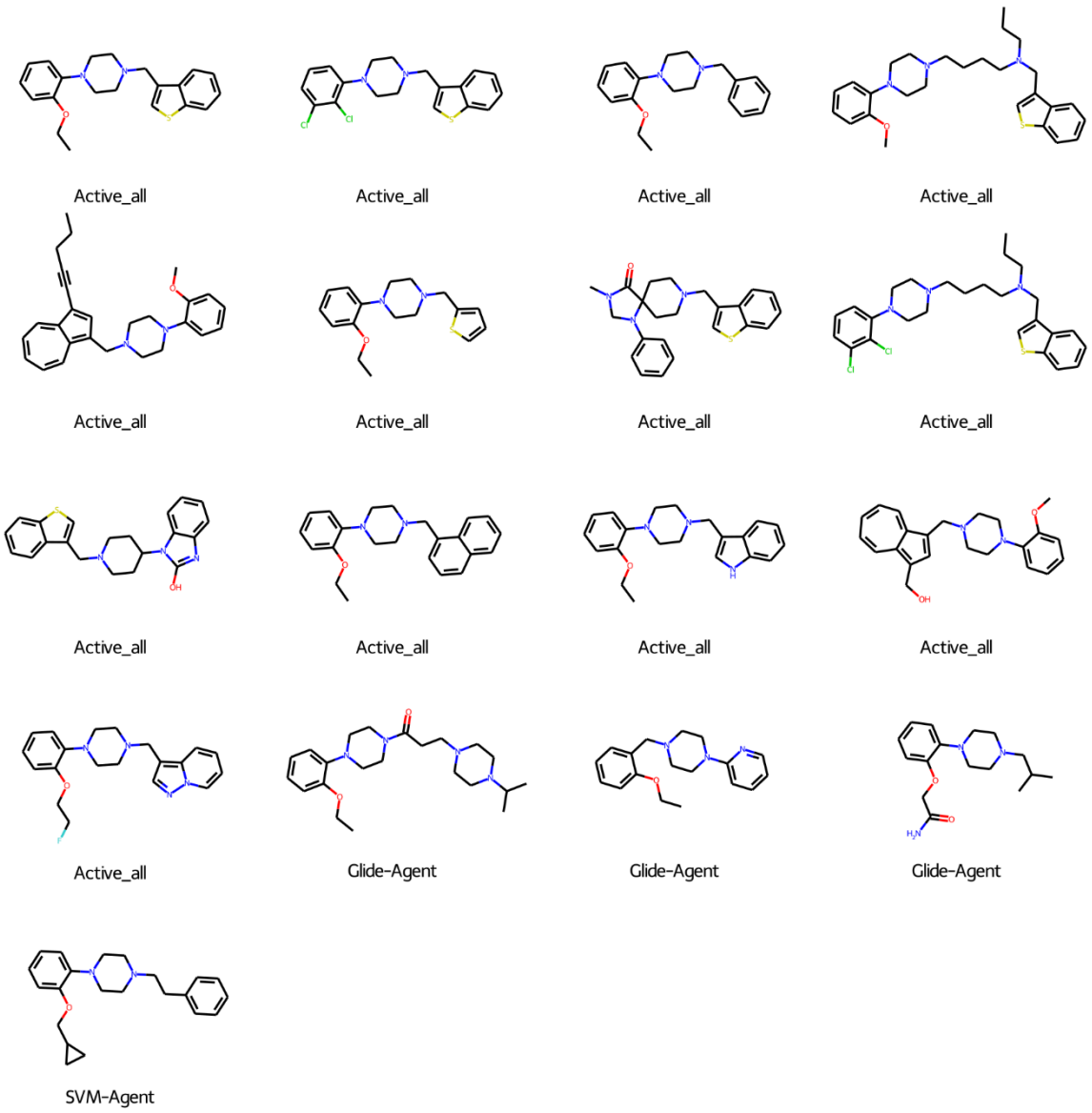


Figure B.1: Example of molecule cluster when defined by whole molecule fingerprints. Molecules clustered using corresponding Morgan fingerprints and a set distance threshold of 0.65. Molecules are somewhat similar although the cluster contains different ring systems and linker lengths.

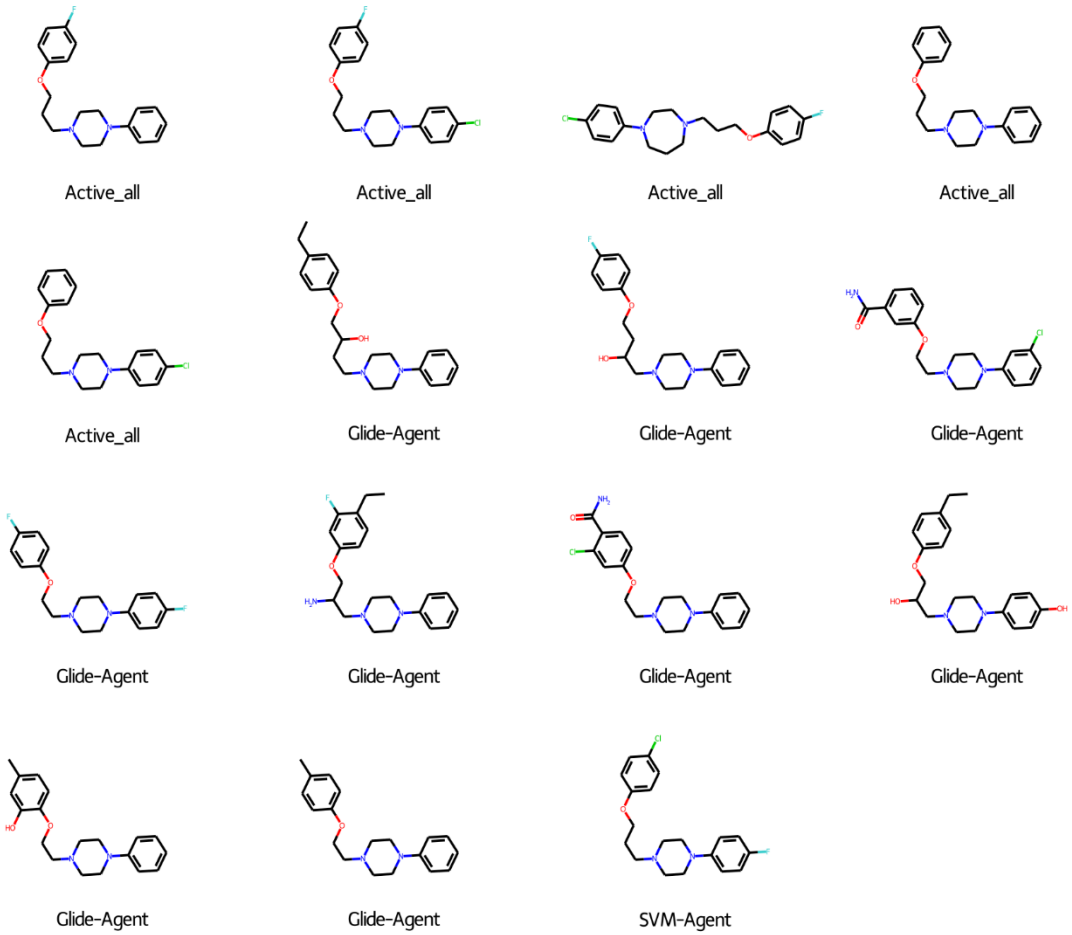


Figure B.2: Example of molecule cluster when defined by Bemis-Murcko scaffold fingerprints. Molecules clustered using corresponding Morgan fingerprints of respective Bemis-Murcko scaffolds with a set distance threshold of 0.2. Molecules are very similar; the linker differs in length by two carbons or less and one central ring is one carbon larger.

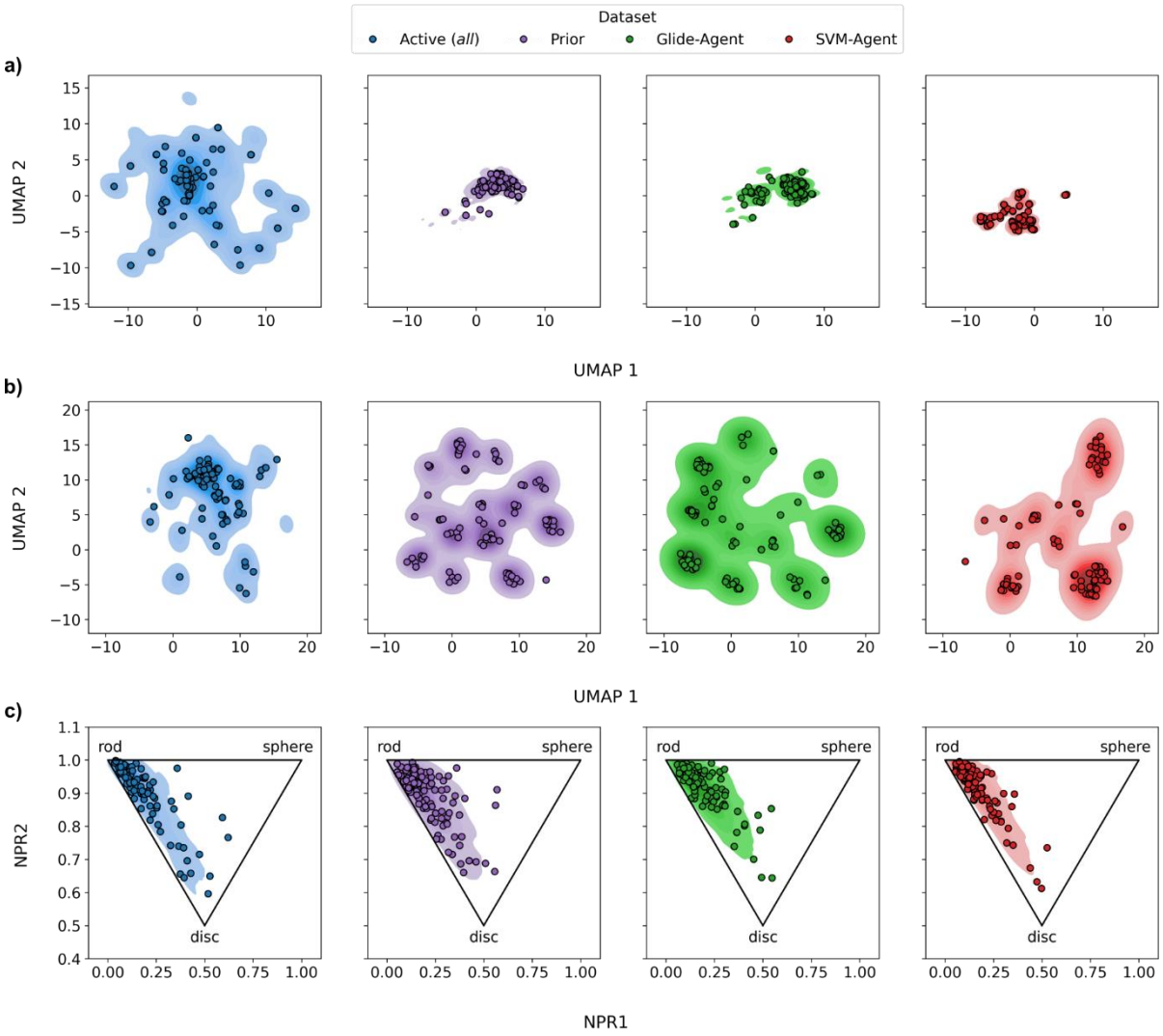


Figure B.3: Chemical space representation of (a) molecular fingerprints and (b) physicochemical descriptors and (c) 3D space via moments of inertia. The plots show the calculated kernel density estimate with 100 randomly drawn samples overlayed. UMAP representation (a-b) was calculated for *all* active DRD2 ligands without filters applied, as well as the chemical structures associated with the Prior, Glide- and SVM-Agents. The Agents occupy complementary regions of topological space (a), physicochemical property space (b) and slightly 3D space (c). It can also be seen that the Glide-Agent better maintains the physicochemical diversity found in the Prior.

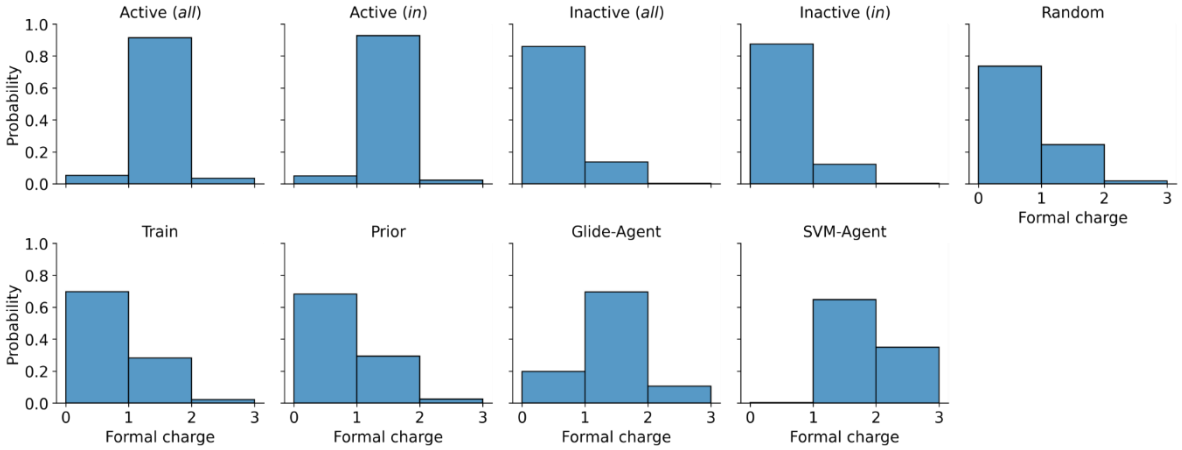


Figure B.4: Formal charge distribution of datasets according to the docking protocol (i.e., protonations states possible at pH 7±1 with the best docking score). The charge distribution observed by the Prior is shifted by the Glide-Agent to closer recapitulate the distribution found in actives (all extracted from ExCAPE-DB), in fact more so than the SVM-Agent which contains more di-cationic molecules. This suggests that the docking scoring function does not over estimate charge contributions.

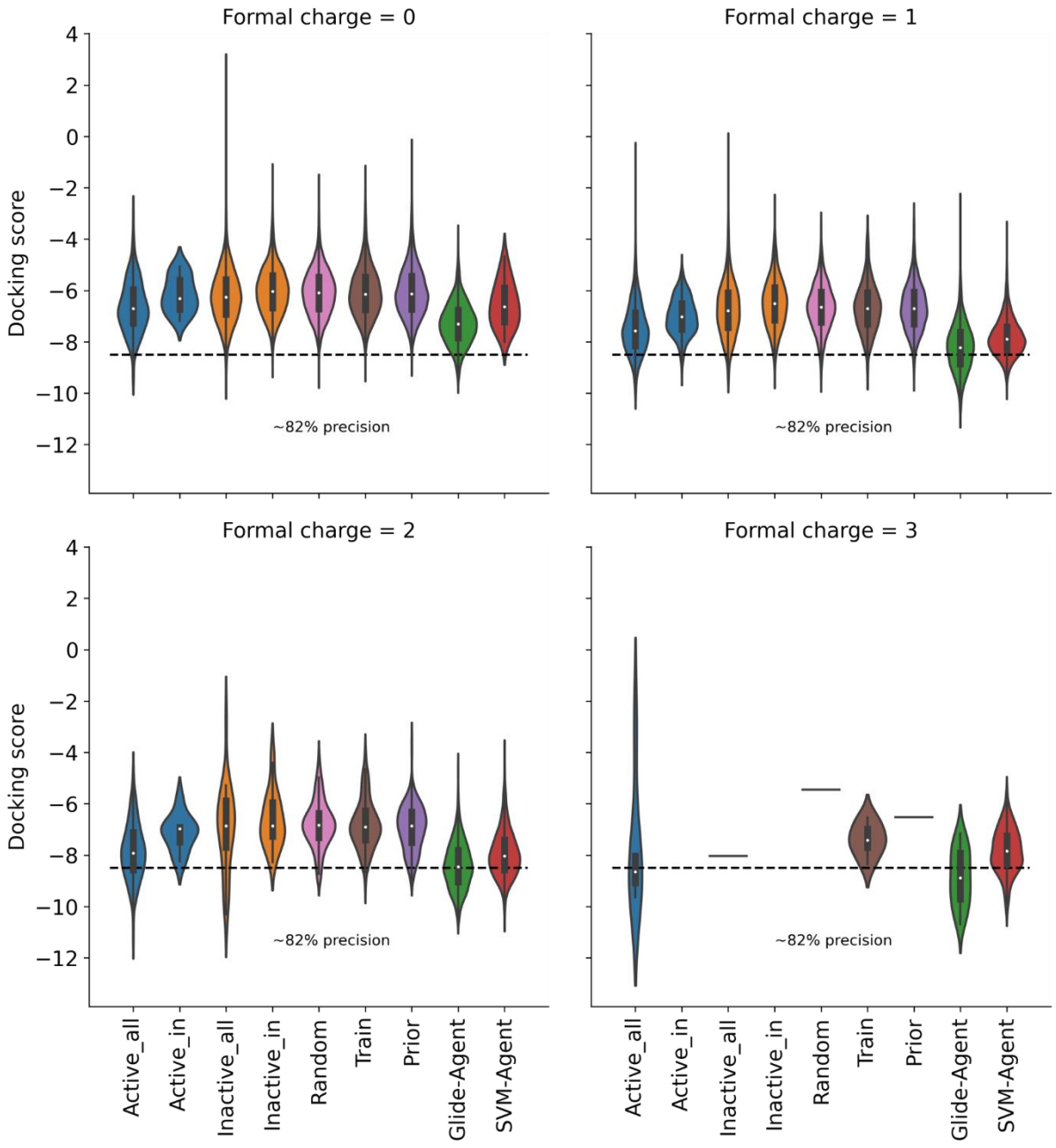


Figure B.5: Docking score distribution of molecules in each dataset split by filtering certain formal charge values. The Glide-Agent provides the most or equal enrichment at all formal charge states.

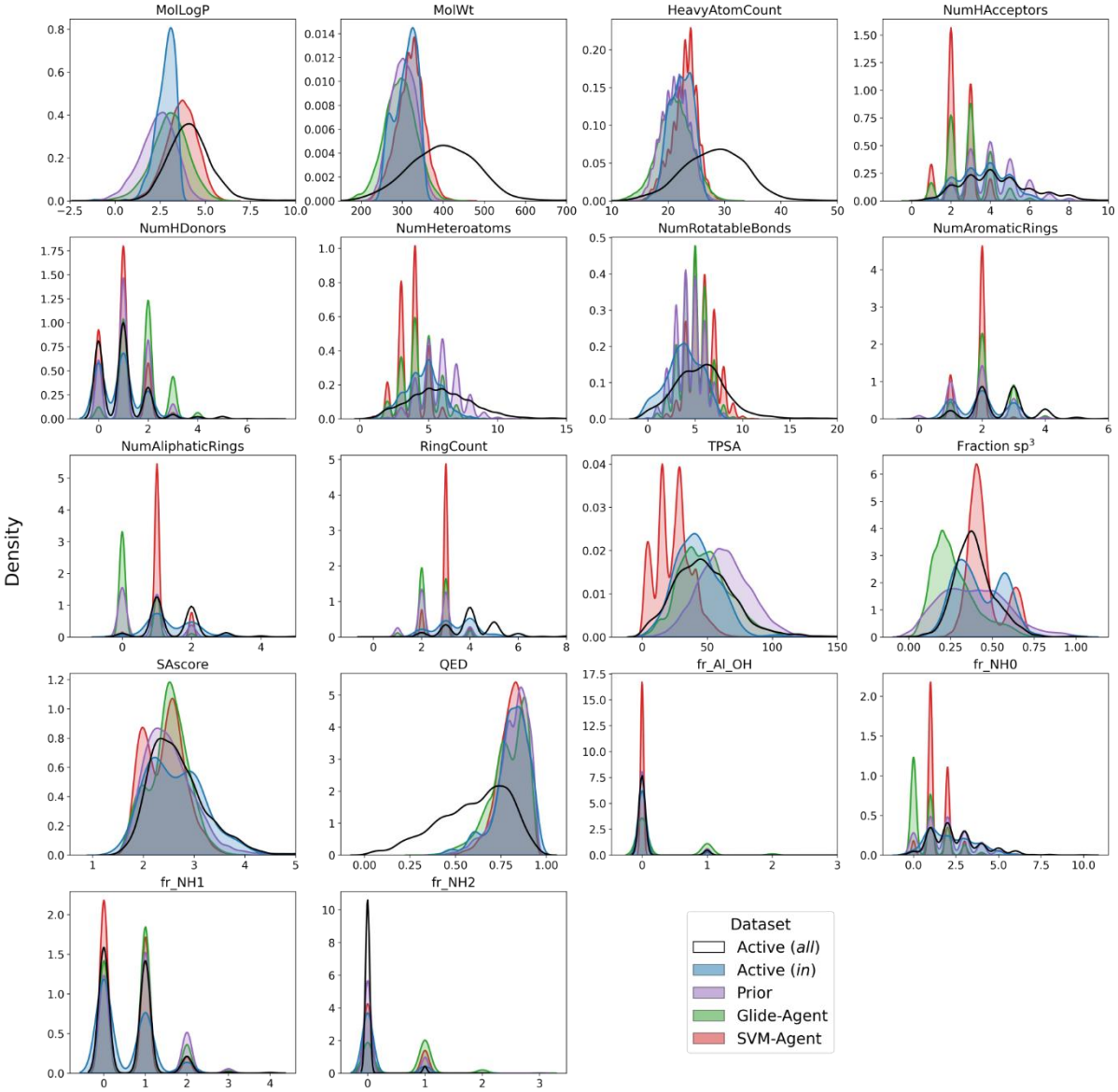


Figure B.6: Kernel density estimates of *de novo* molecule physicochemical properties. Including SAscore, QED, number of aliphatic hydroxyl groups (fr_Al_OH) and number of tertiary, secondary and primary amines (fr_NH0, fr_NH1, fr_NH2). Of note, the Glide-Agent molecular weight distribution diverges away from the Prior and DRD2 active molecules.

Appendix C

Table C.1: Number of molecules downloaded from ExCAPE-DB and those docked against targets used for evaluating Augmented Hill-Climb to assess retrospective performance.

	ExCAPE-DB		**Docked**	
	Active	Inactive	Active	Inactive
DRD2	4613	343076	3734	9538
OPRM1	3128	2786	3125	2573
AGTR1	671	558	270	516
OX1R	681	322795	564	9860

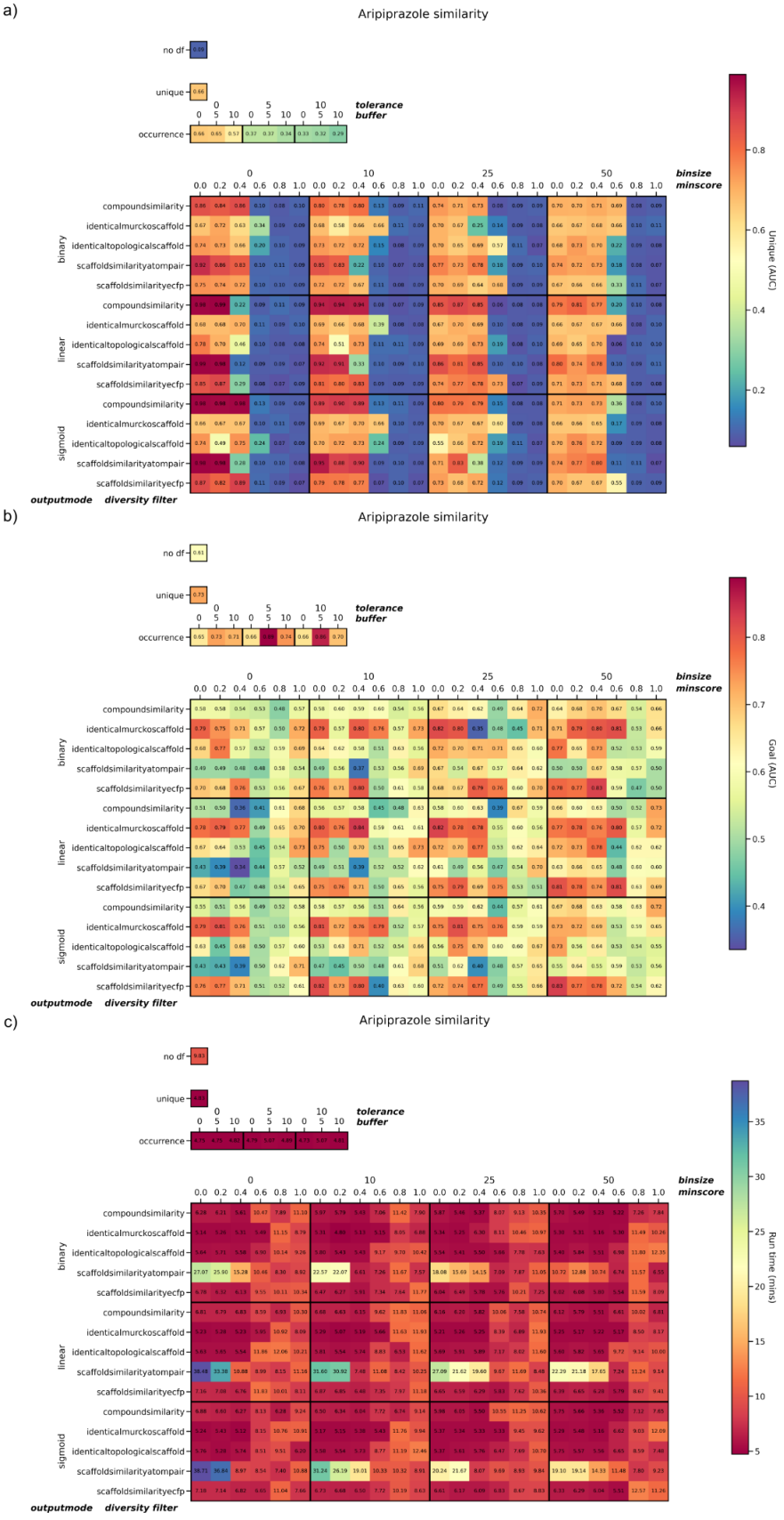


Figure C.1: AHC optimization of Aripiprazole similarity task with different diversity filters and their parameters. Three endpoints are measured: (**a**) unique area under the curve (AUC), (**b**) goal AUC (**c**) and run time.

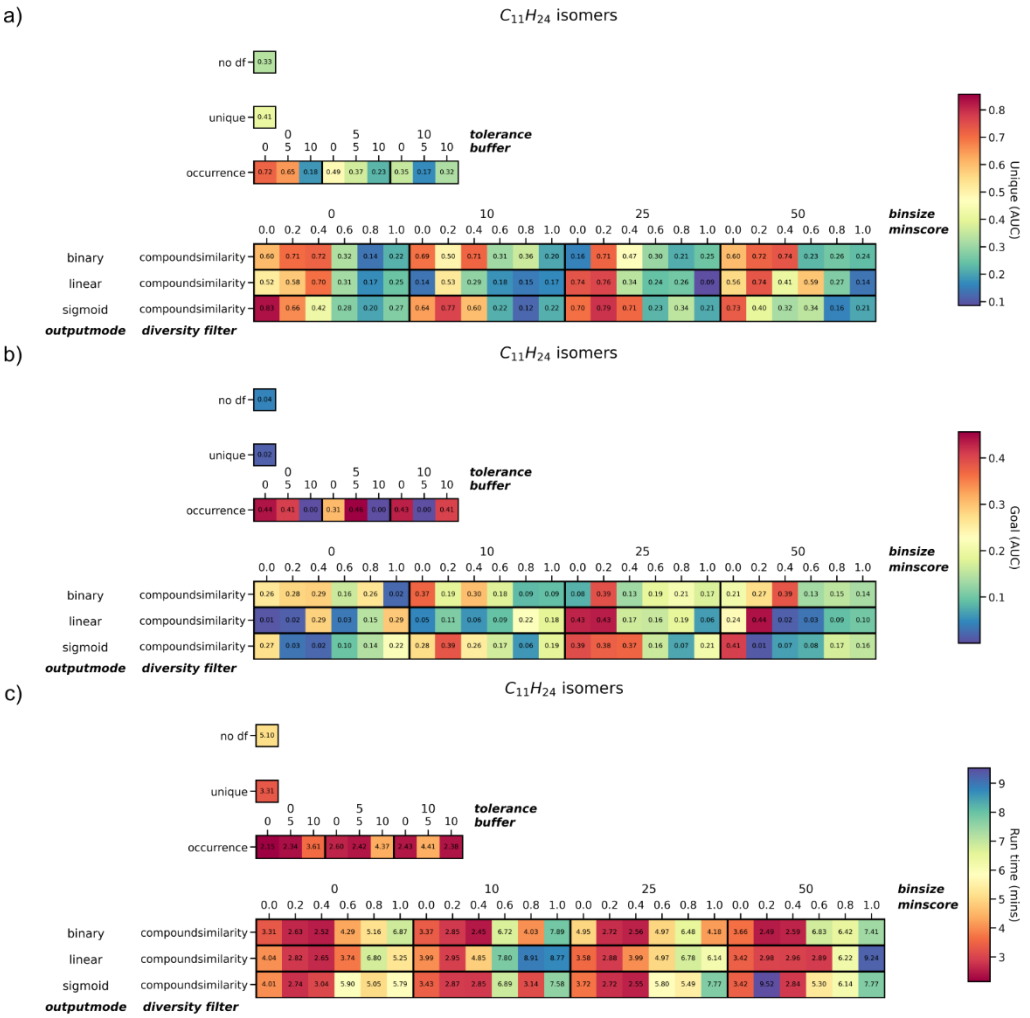


Figure C.2: AHC optimization of $C_{11}H_{24}$ isomers task with different diversity filters and their parameters. Three endpoints are measured: (a) unique area under the curve (AUC), (b) goal AUC (c) and run time. Note: scaffold-based diversity filters aren't shown as $C_{11}H_{24}$ isomers cannot form rings and satisfy the molecular formula and so no scaffolds will be detected.

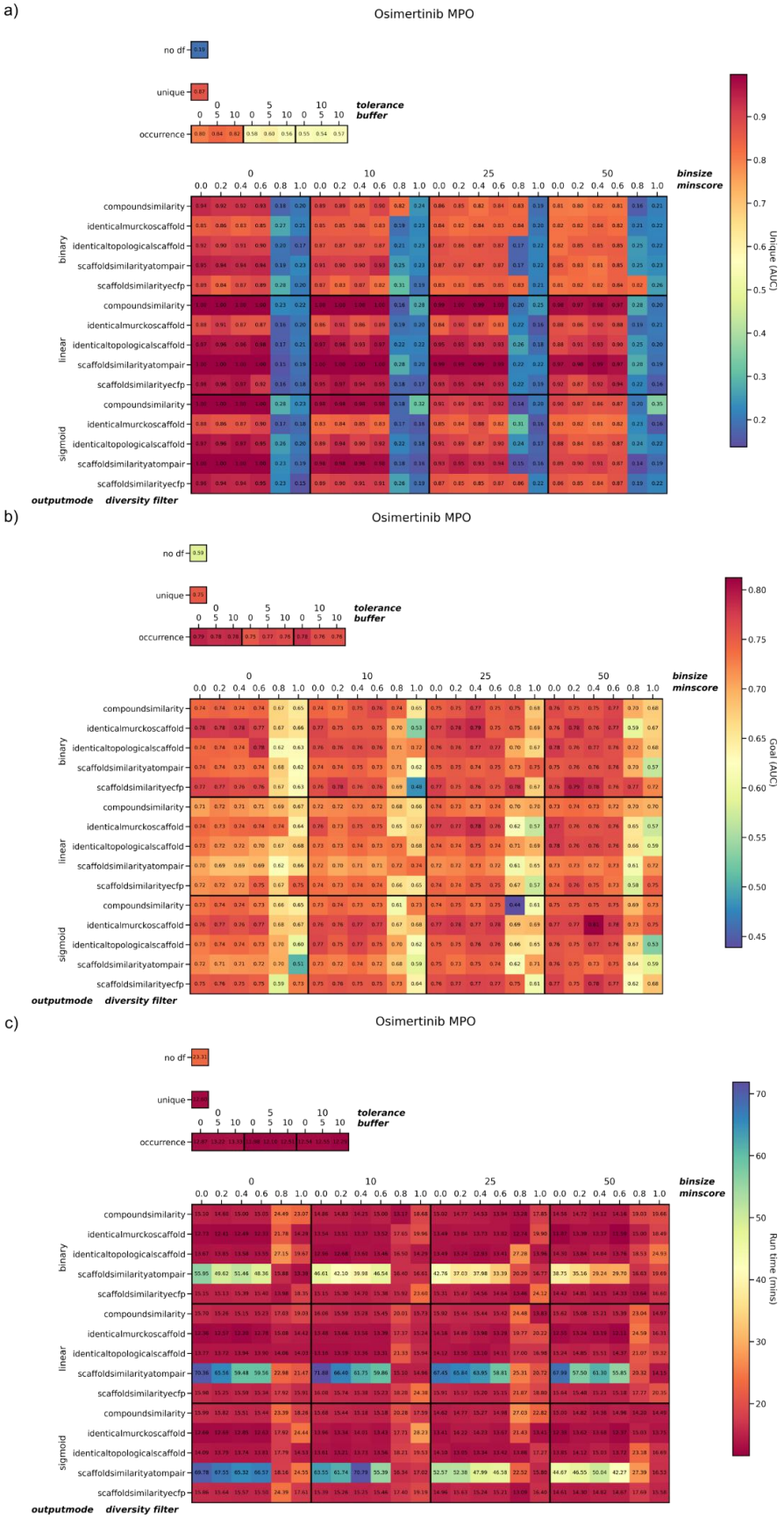


Figure C.3: AHC optimization of Osimertinib MPO task with different diversity filters and their parameters. Three endpoints are measured: (a) unique area under the curve (AUC), (b) goal AUC (c) and run time.

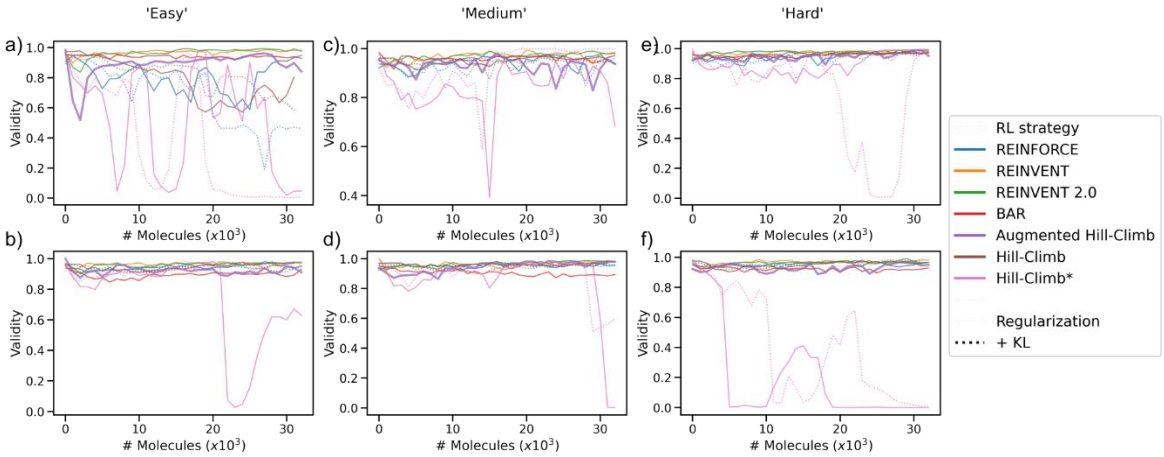


Figure C.4: Validity for objective optimization using different RL strategies. HC* suffers from a drop in validity.

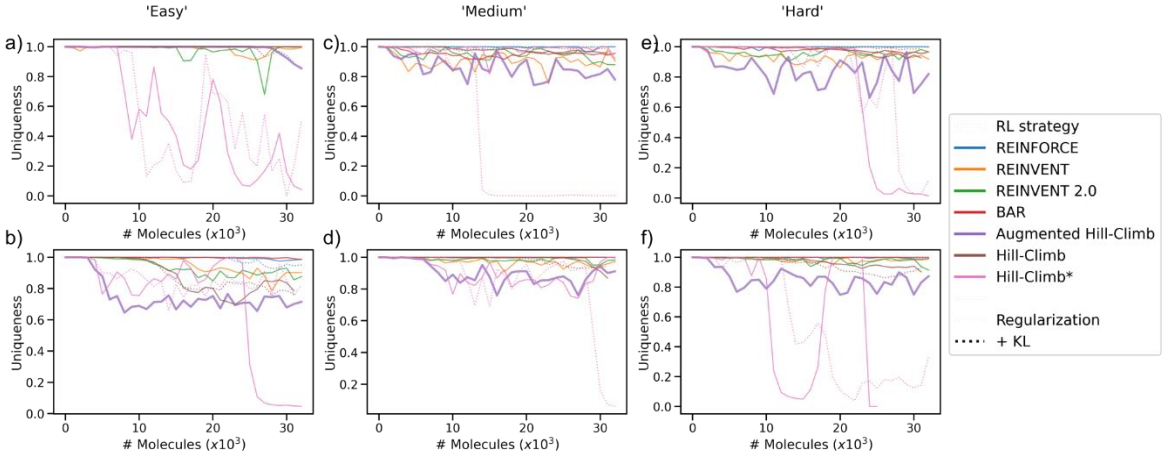


Figure C.5: Uniqueness for objective optimization using different RL strategies. HC* suffers from a drop in uniqueness.

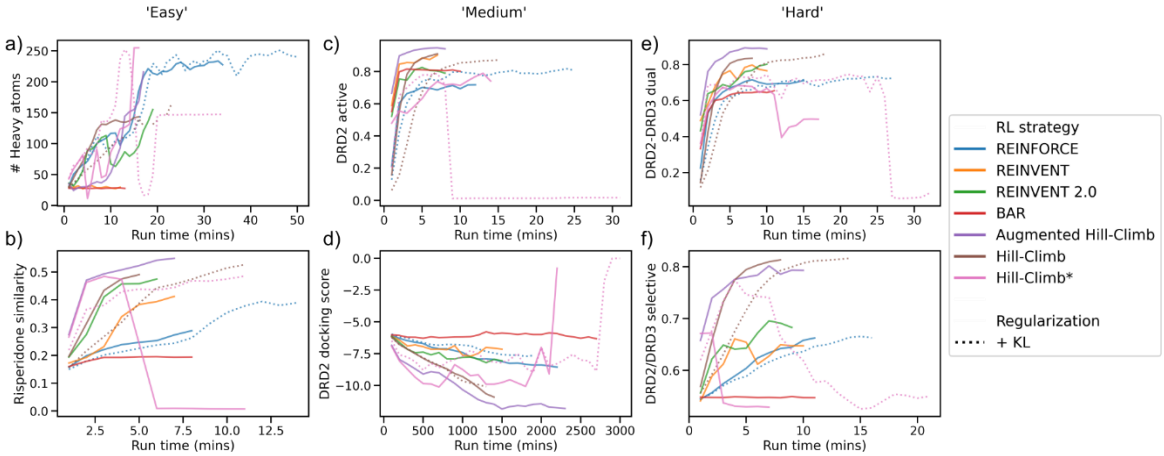


Figure C.6: Wall time for objective optimization using different RL strategies. Run using an AMD Threadripper 1920x CPU and Nvidia GeForce RTX 2060 super GPU. The docking tasks were parallelized over 10 CPU cores while all other tasks used only 1 CPU core.

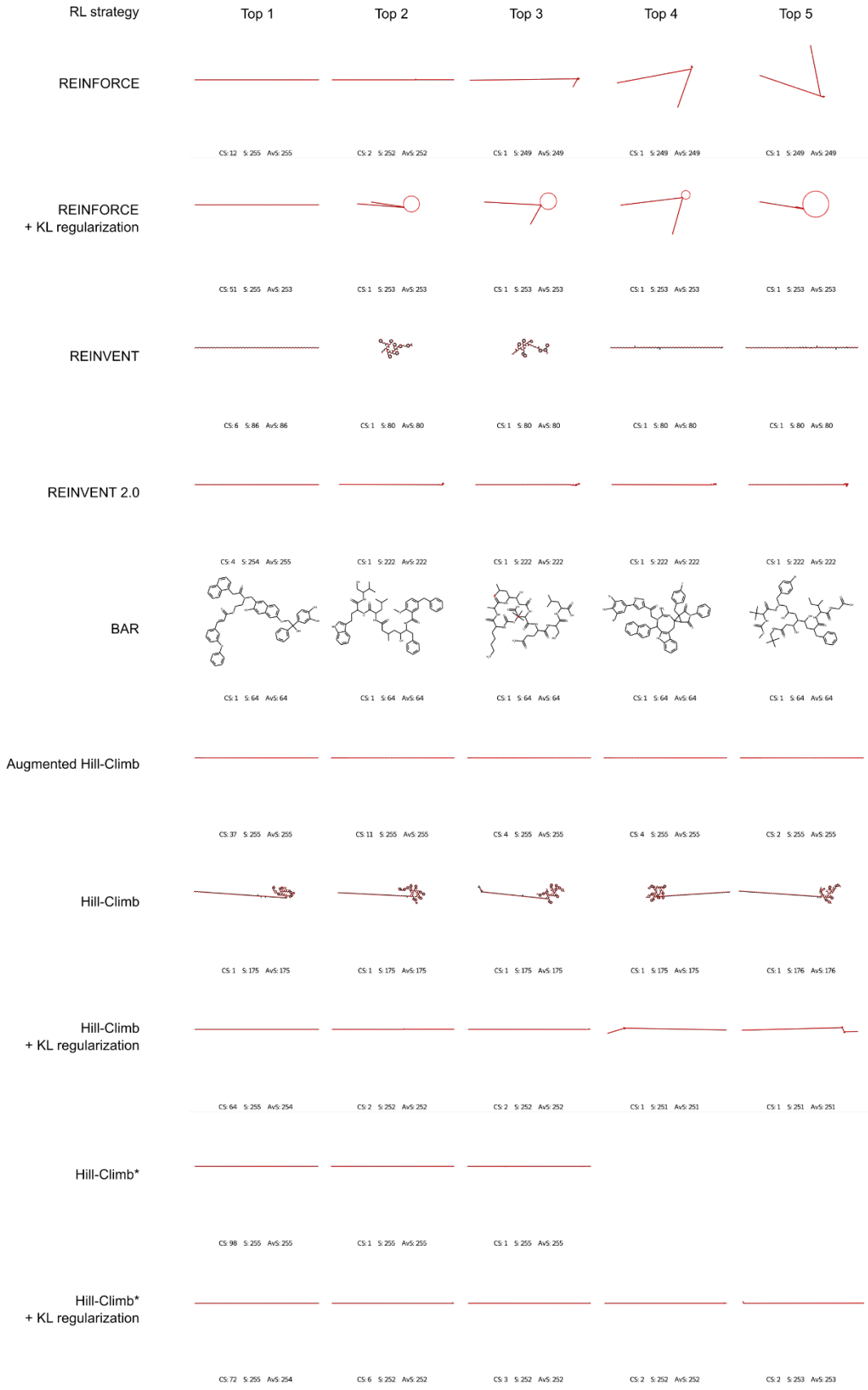


Figure C.7: Centroid of the top 5 largest clusters for the top 100 molecules in the heavy atom task for different RL strategies. Cluster size (CS), centroid score (S) and the average cluster score (AvS) is annotated below. Long chains can be seen due to the nature of the task increasing the number of heavy atoms.

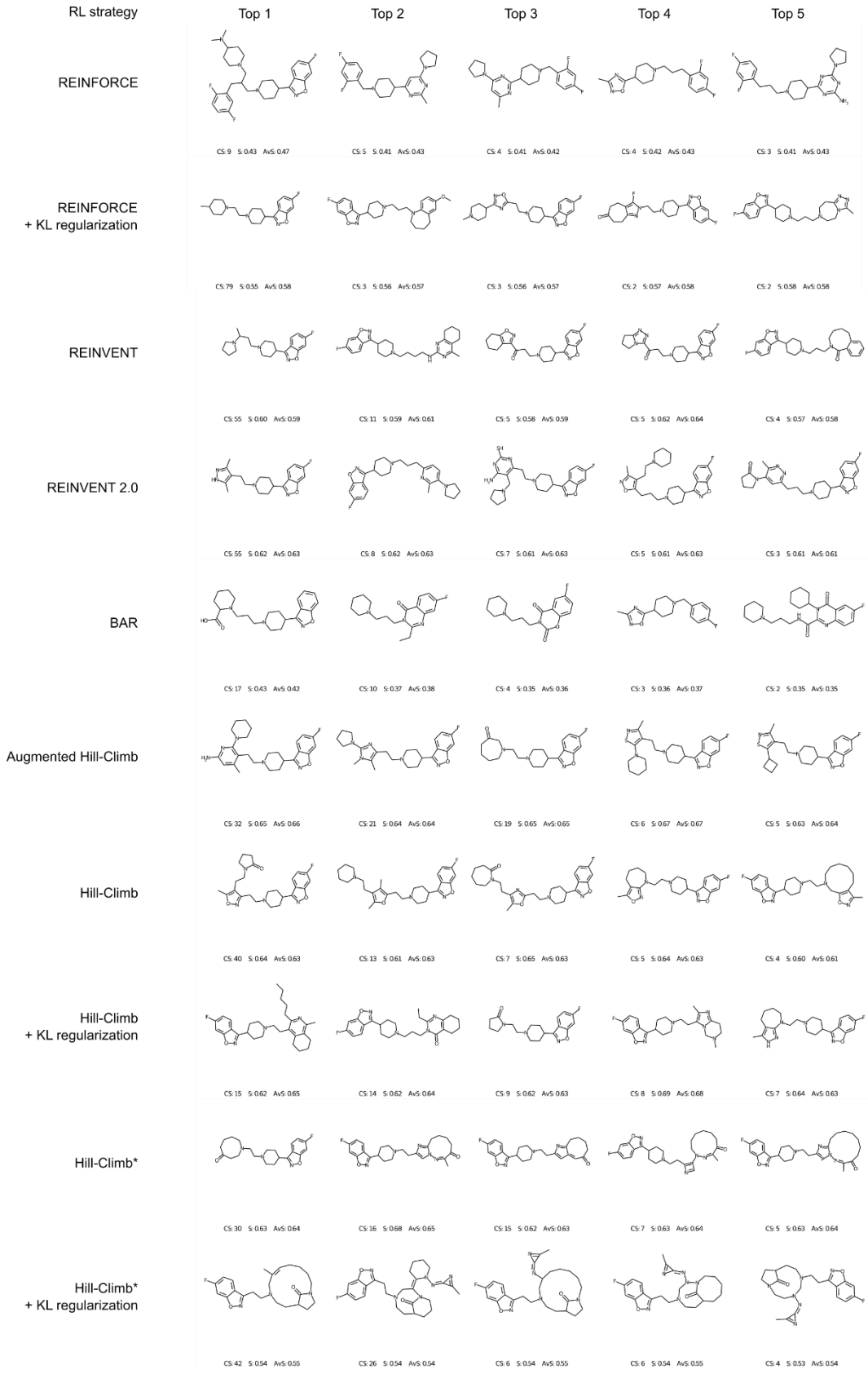


Figure C.8: Centroid of the top 5 largest clusters for the top 100 molecules in the Risperidone similarity task for different RL strategies. Cluster size (CS), centroid score (S) and the average cluster score (AvS) is annotated below.

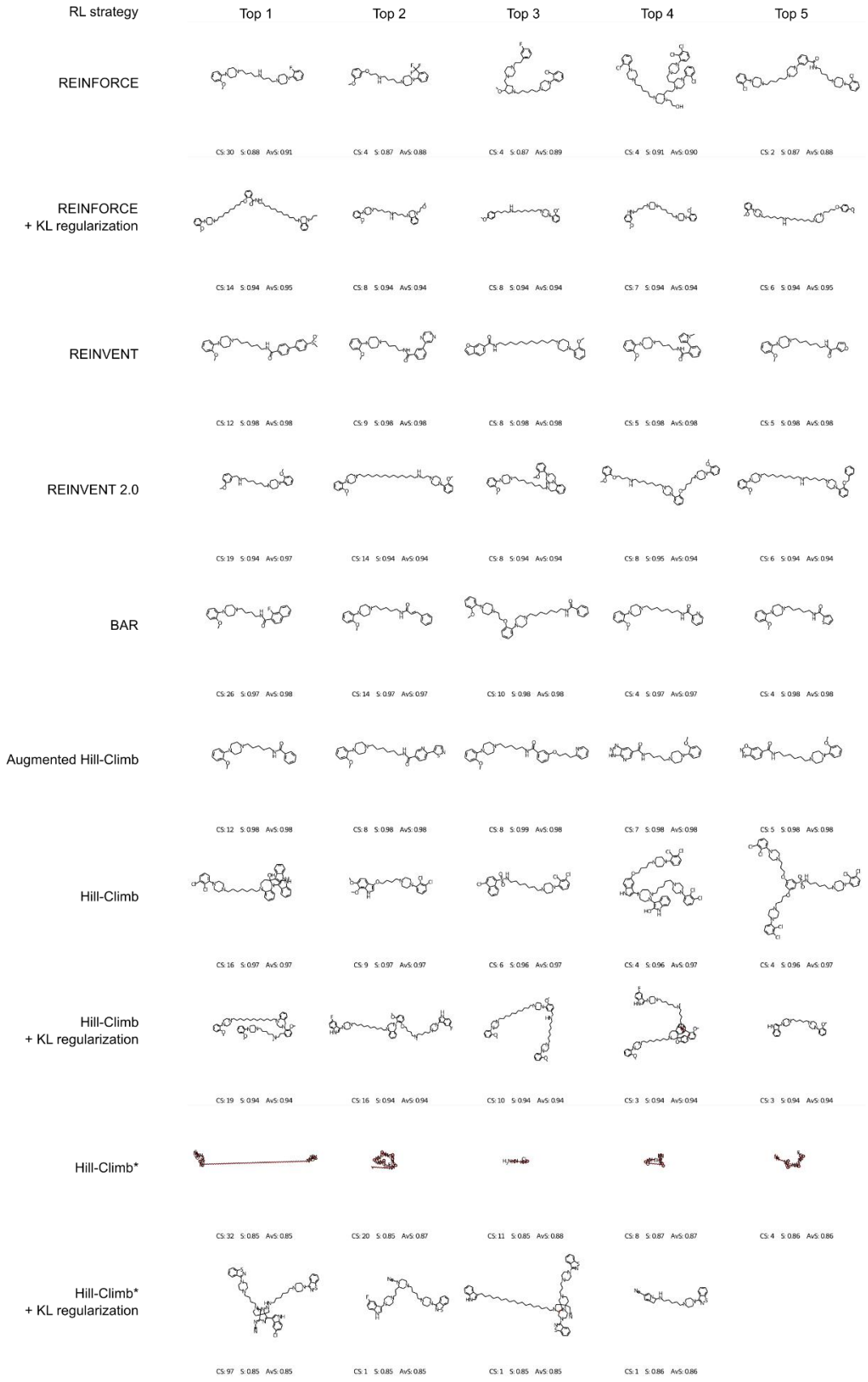


Figure C.9: Centroid of the top 5 largest clusters for the top 100 molecules in the DRD2 activity task for different RL strategies. Cluster size (CS), centroid score (S) and the average cluster score (AvS) is annotated below.

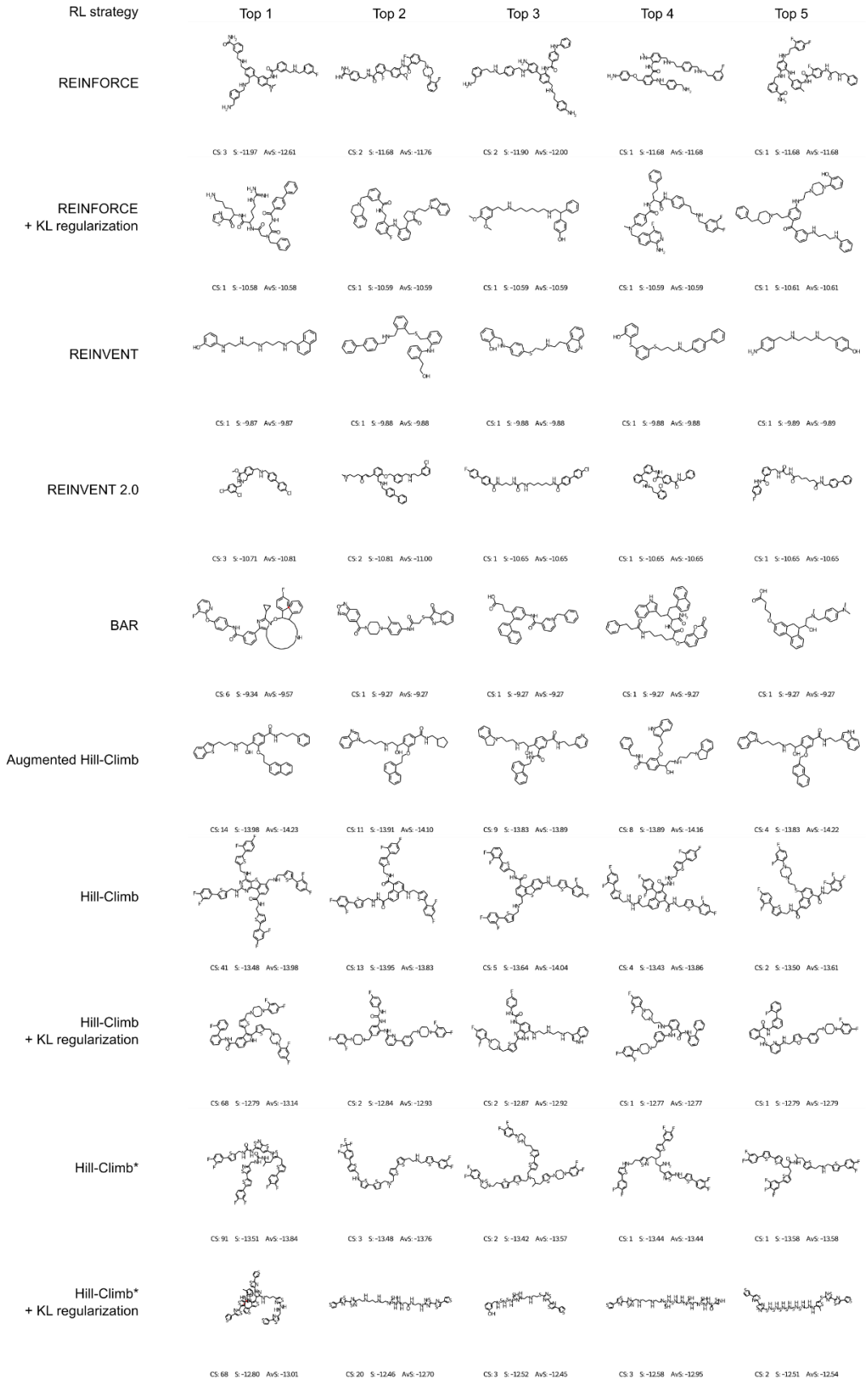


Figure C.10: Centroid of the top 5 largest clusters for the top 100 molecules in the DRD2 docking score task for different RL strategies. Cluster size (CS), centroid score (S) and the average cluster score (AvS) is annotated below.

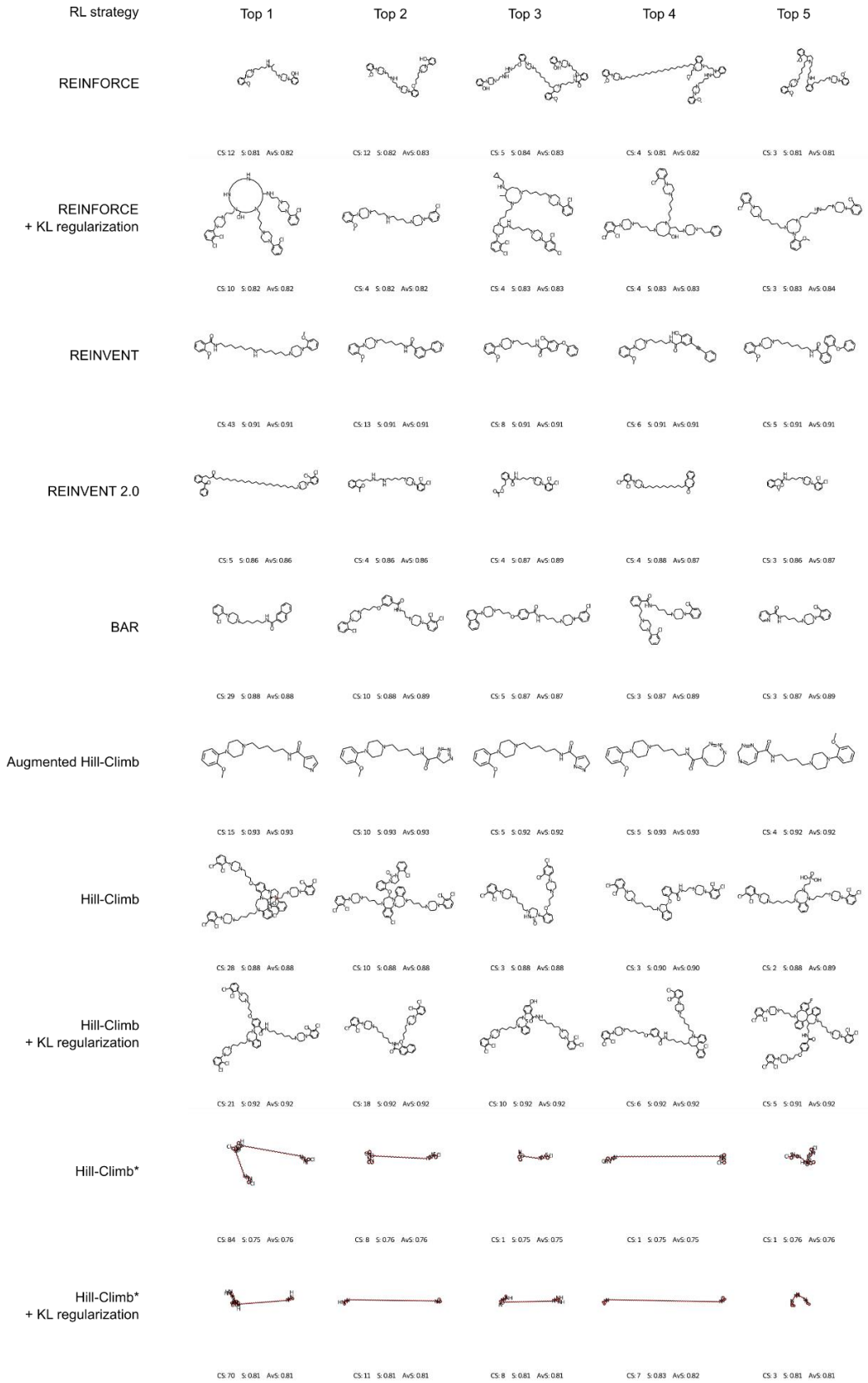


Figure C.11: Centroid of the top 5 largest clusters for the top 100 molecules in the DRD2-DRD3 QSAR dual predicted probability of activity task for different RL strategies. Cluster size (CS), centroid score (S) and the average cluster score (AvS) is annotated below. Note where AHC generates cationic species due to imperfections in the neutralization of training data resulting in charge symbols in the RNN vocabulary.

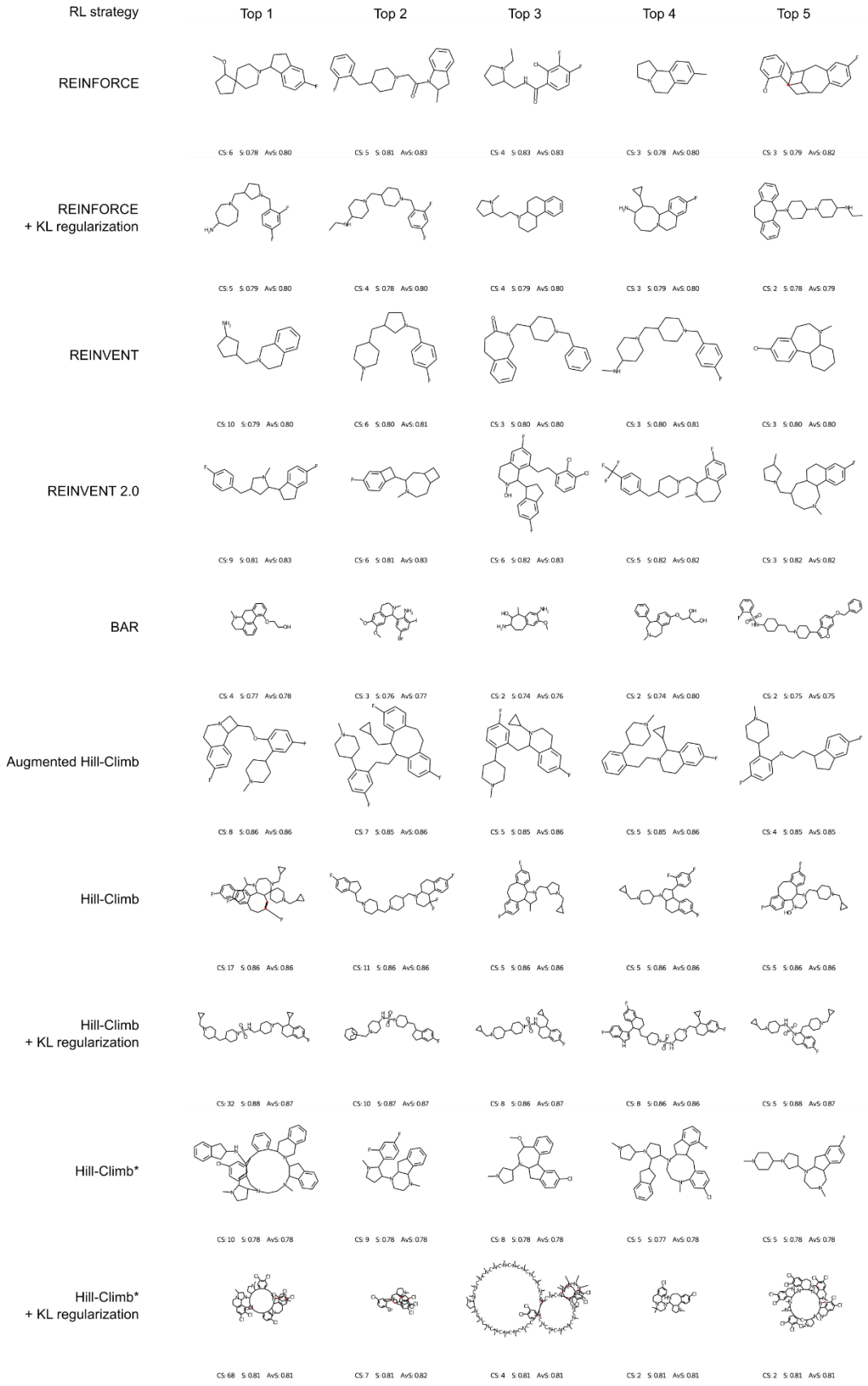


Figure C.12: Centroid of the top 5 largest clusters for the top 100 molecules in the DRD2/DRD3 QSAR predicted probability of selective activity task for different RL strategies. Cluster size (CS), centroid score (S) and the average cluster score (AvS) is annotated below.

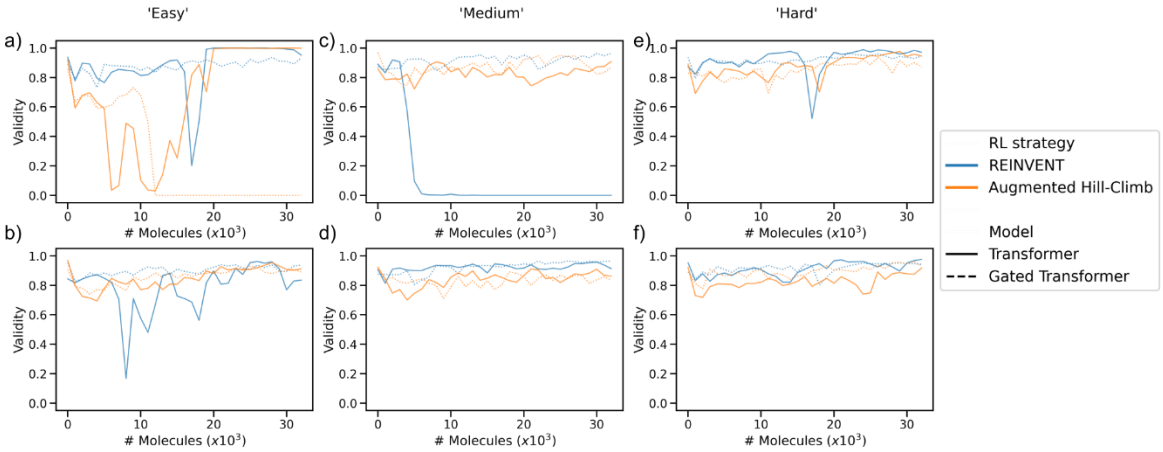


Figure C.13: Validity for objective optimization using REINVENT and Augmented Hill-Climb with a transformer or gated transformer model.

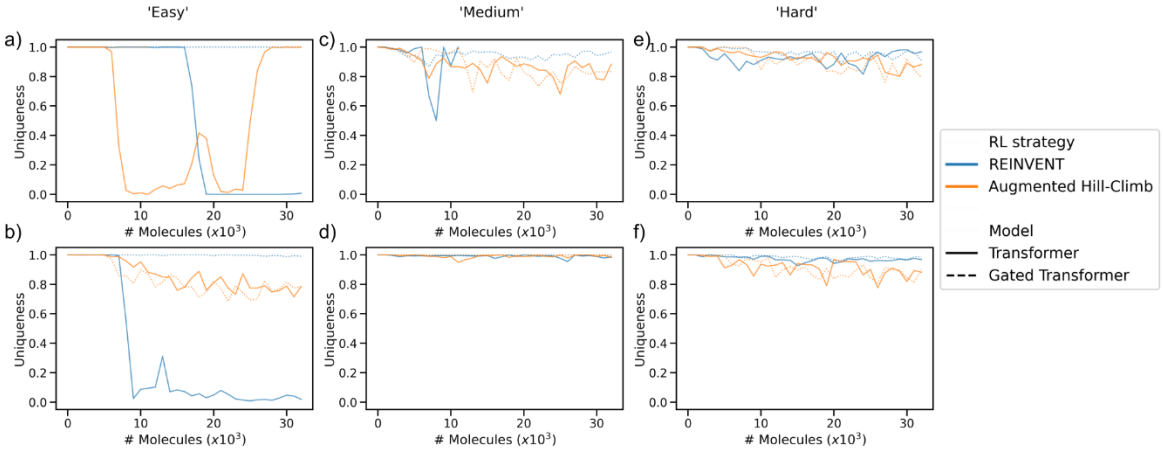


Figure C.14: Uniqueness for objective optimization using REINVENT and Augmented Hill-Climb with a transformer or gated transformer model. Transformer model is more prone to undergoing a drop in uniqueness.

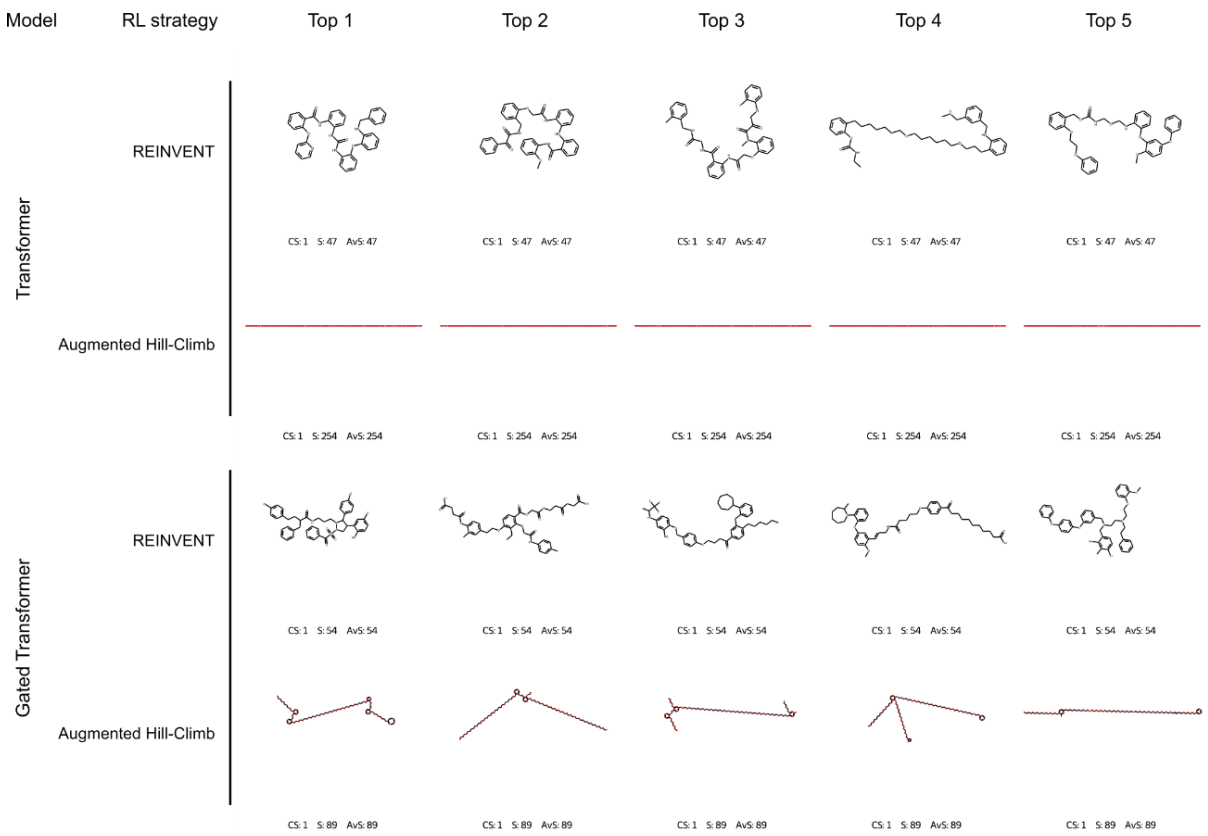


Figure C.15: Centroid of the top 5 largest clusters for the top 100 molecules in the heavy atom task for transformer models. Cluster size (CS), centroid score (S) and the average cluster score (AvS) is annotated below.

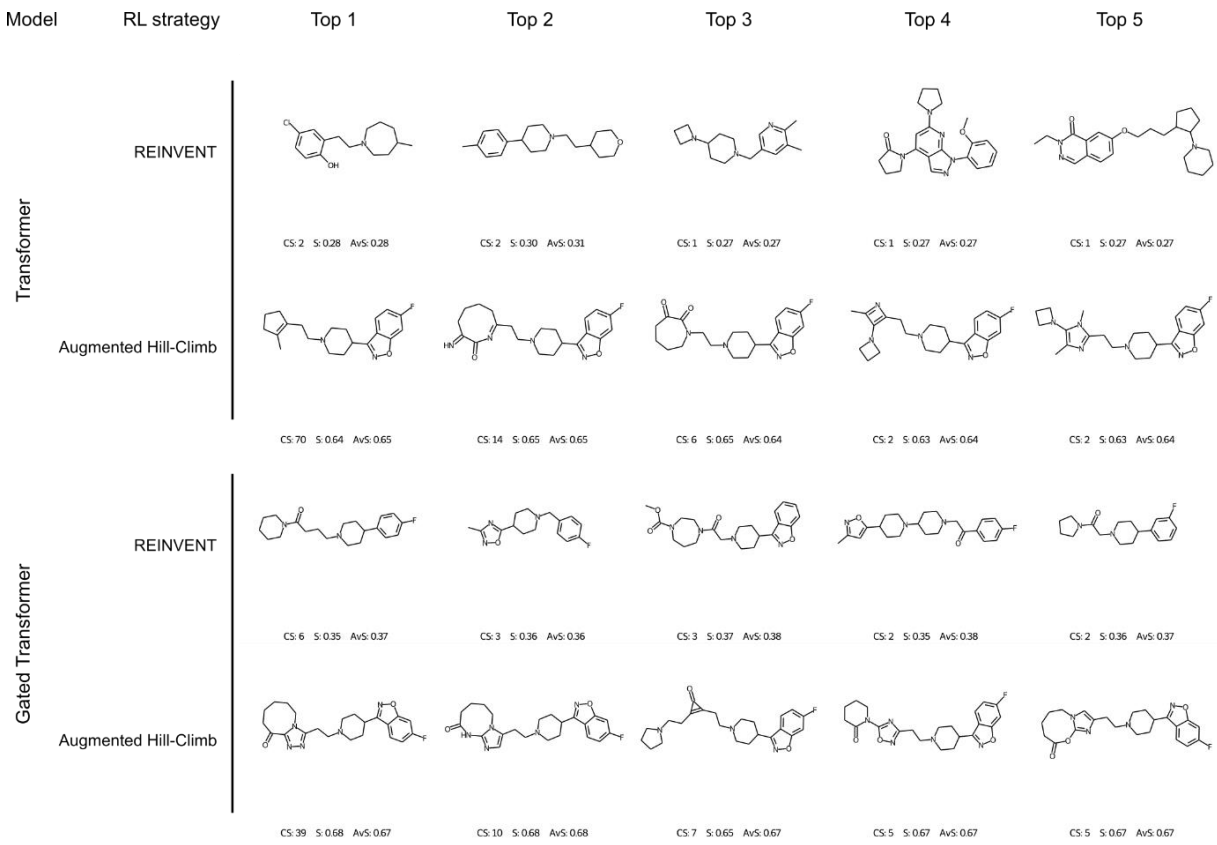


Figure C.16: Centroid of the top 5 largest clusters for the top 100 molecules in the Risperidone similarity task for transformer models. Cluster size (CS), centroid score (S) and the average cluster score (AvS) is annotated below.

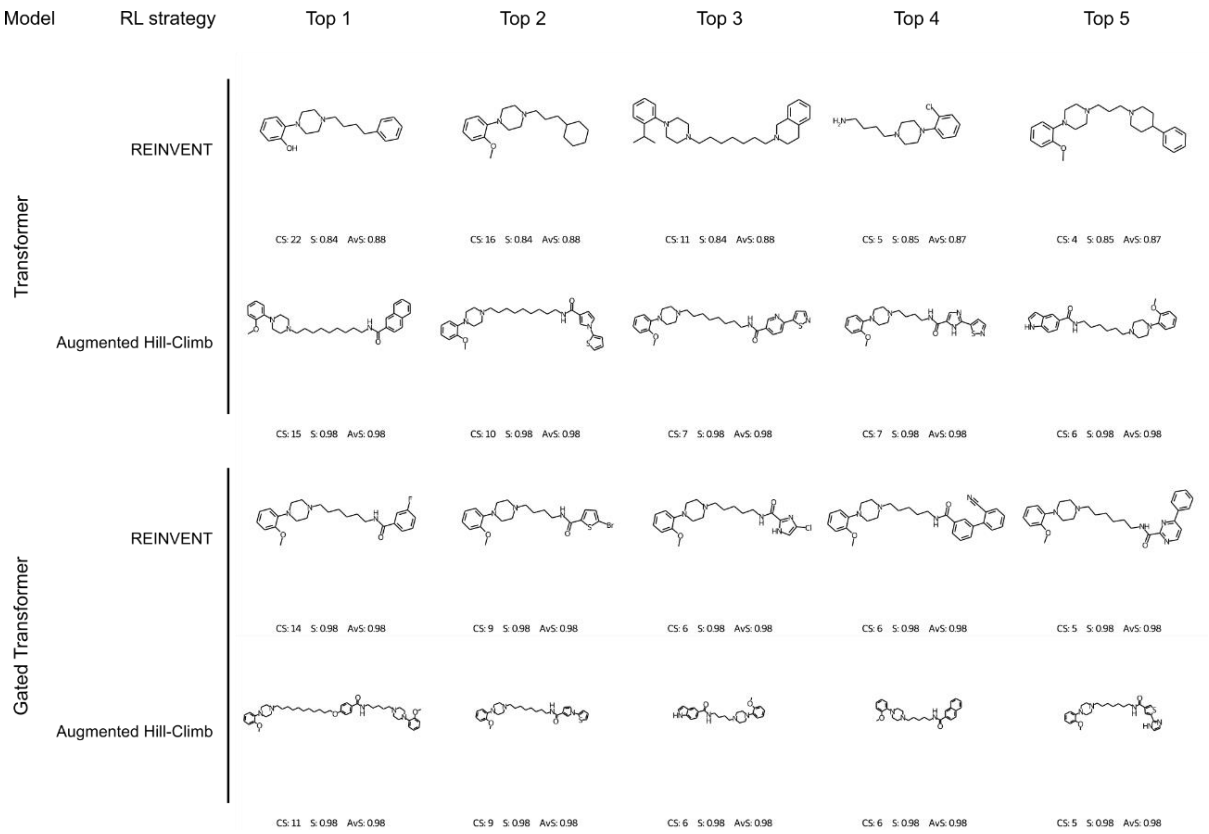


Figure C.17: Centroid of the top 5 largest clusters for the top 100 molecules in the DRD2 activity task for transformer models. Cluster size (CS), centroid score (S) and the average cluster score (AvS) is annotated below.

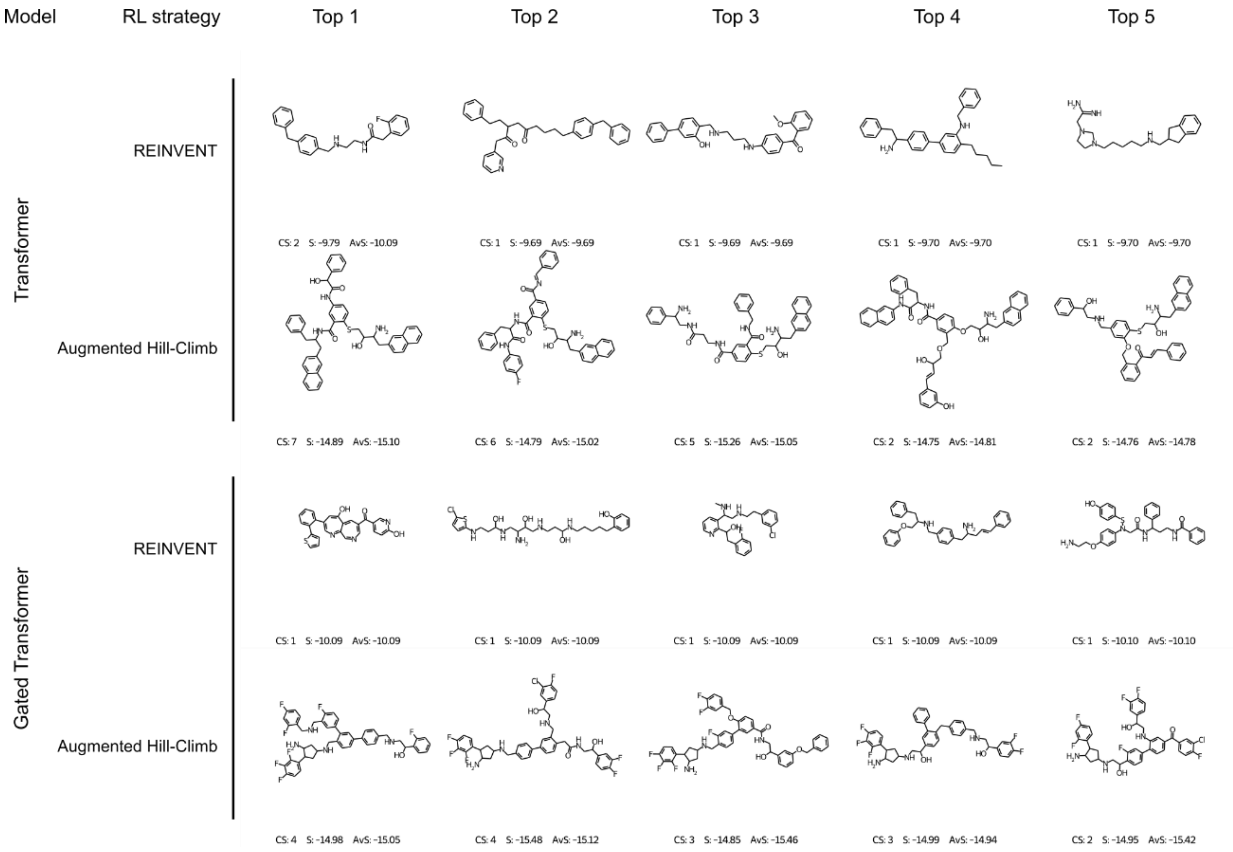


Figure C.18: Centroid of the top 5 largest clusters for the top 100 molecules in the DRD2 docking score task for transformer models. Cluster size (CS), centroid score (S) and the average cluster score (AvS) is annotated below.

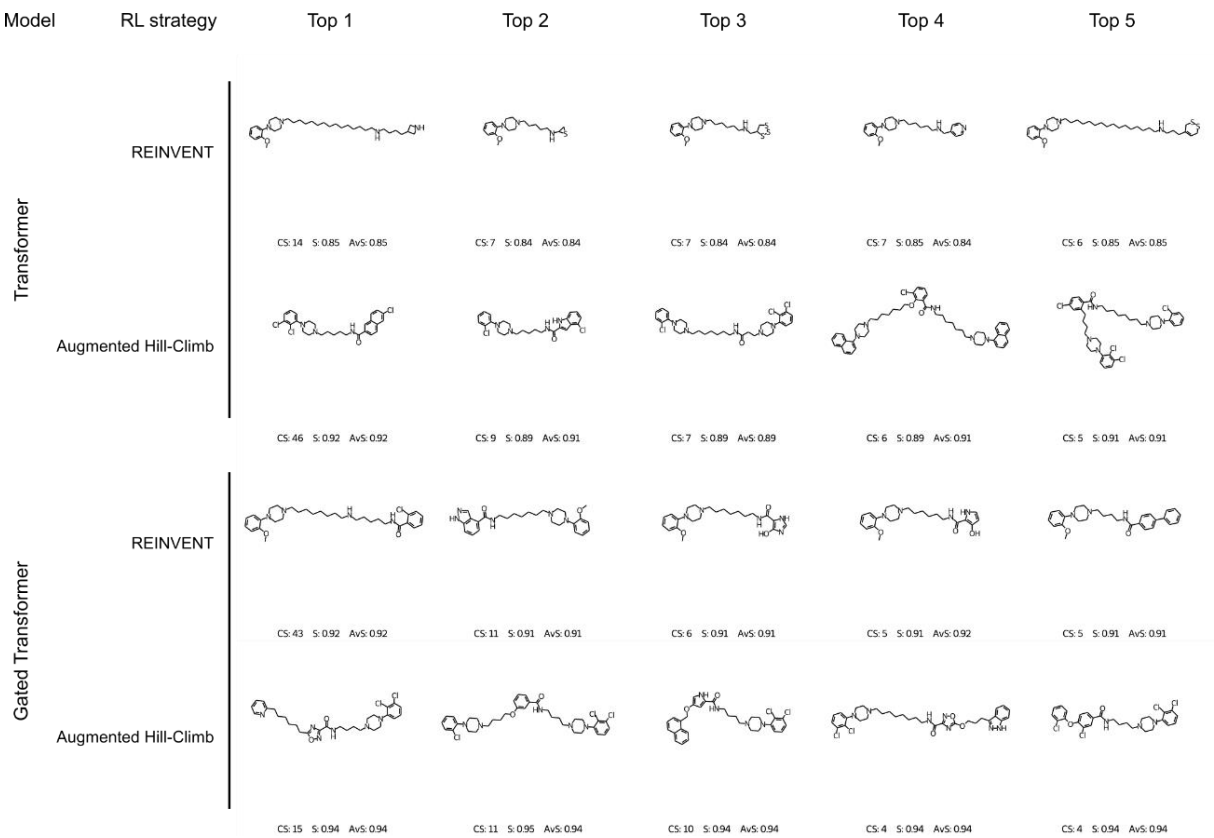


Figure C.19: Centroid of the top 5 largest clusters for the top 100 molecules in the DRD2-DRD3 QSAR dual predicted probability of activity task for transformer models. Cluster size (CS), centroid score (S) and the average cluster score (AvS) is annotated below.

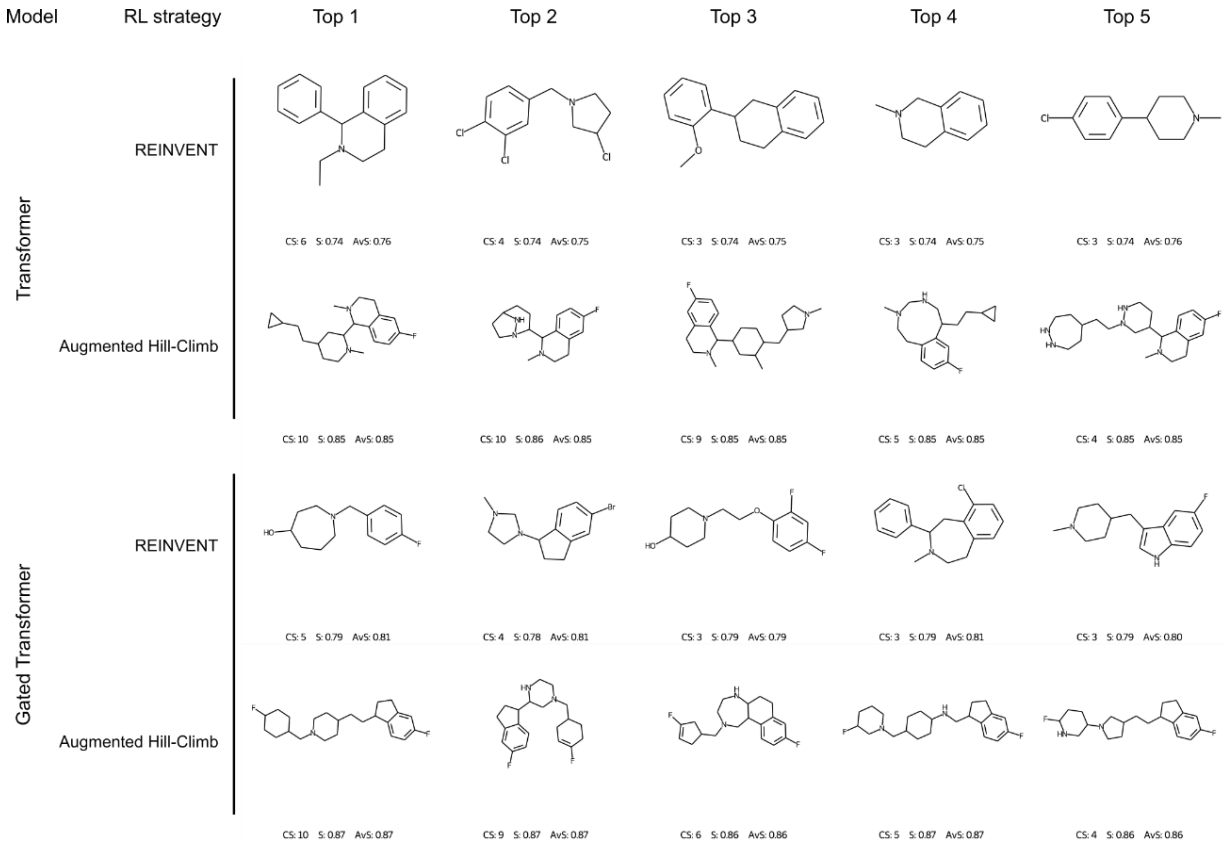
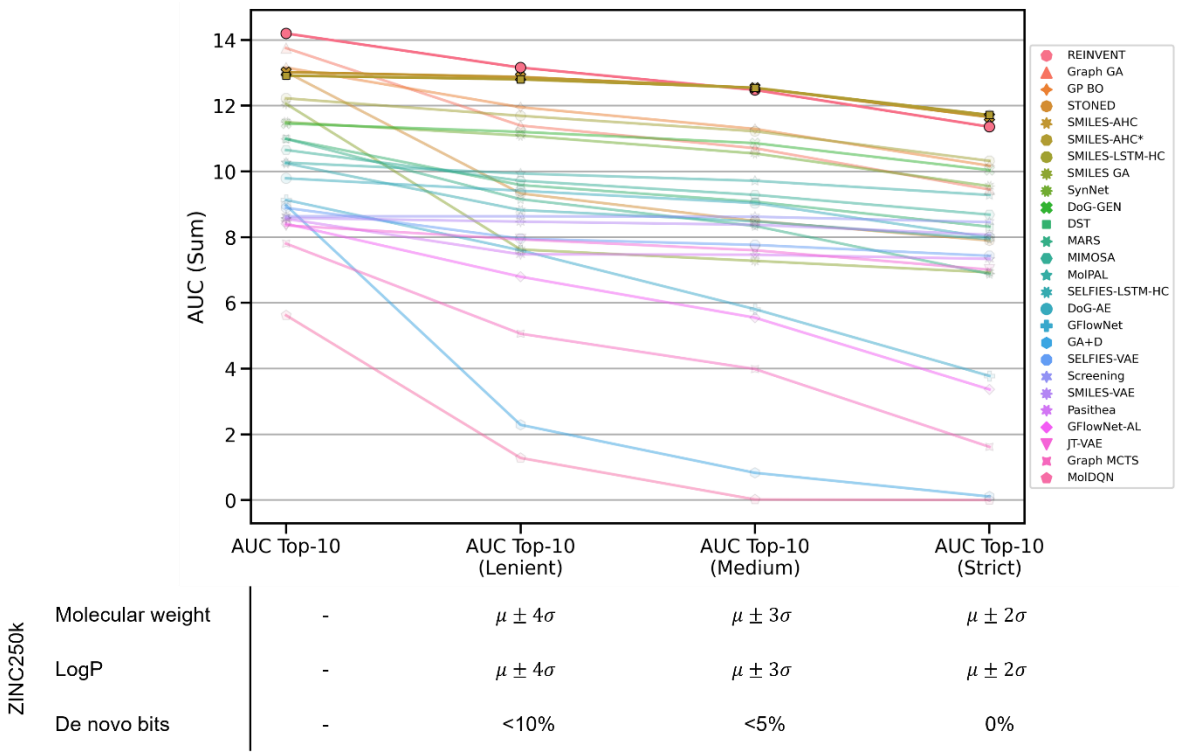


Figure C.20: Centroid of the top 5 largest clusters for the top 100 molecules in the DRD2/DRD3 QSAR predicted probability of selective activity task for different transformer models. Cluster size (CS), centroid score (S) and the average cluster score (AvS) is annotated below.



ZINC250k				
Molecular weight	-	$\mu \pm 4\sigma$	$\mu \pm 3\sigma$	$\mu \pm 2\sigma$
LogP	-	$\mu \pm 4\sigma$	$\mu \pm 3\sigma$	$\mu \pm 2\sigma$
De novo bits	-	<10%	<5%	0%

Figure C.21: Performance of generative models on PMO benchmark at different levels of chemical constraints relative to the training dataset ZINC250k.

References

1. DiMasi, J. A., Grabowski, H. G. & Hansen, R. W. Innovation in the pharmaceutical industry: New estimates of R&D costs. *J. Health Econ.* **47**, 20–33 (2016).
2. Mak, K. K. & Pichika, M. R. Artificial intelligence in drug development: present status and future prospects. *Drug Discov. Today* **24**, 773–780 (2019).
3. MacArron, R. *et al.* Impact of high-throughput screening in biomedical research. *Nat. Rev. Drug Discov. 2011 103* **10**, 188–195 (2011).
4. Goodwin, S., Shahtahmassebi, G. & Hanley, Q. S. Statistical models for identifying frequent hitters in high throughput screening. *Sci. Reports 2020 101* **10**, 1–13 (2020).
5. Volochnyuk, D. M. *et al.* Evolution of commercially available compounds for HTS. *Drug Discov. Today* **24**, 390–402 (2019).
6. Schneider, P. & Schneider, G. Privileged Structures Revisited. *Angew. Chemie Int. Ed.* **56**, 7971–7974 (2017).
7. Baell, J. B. & Holloway, G. A. 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 (2010).
8. Makarenkov, V. *et al.* An efficient method for the detection and elimination of systematic error in high-throughput screening. *Bioinformatics* **23**, 1648–1657 (2007).
9. Lipinski, C. A. Lead- and drug-like compounds: the rule-of-five revolution. *Drug Discov. Today Technol.* **1**, 337–341 (2004).
10. Polishchuk, P. G., Madzhidov, T. I. & Varnek, A. Estimation of the size of drug-like chemical space based on GDB-17 data. *J. Comput. Aided. Mol. Des.* **27**, 675–679 (2013).
11. Shoichet, B. K. Virtual screening of chemical libraries. *Nature* **432**, 862–865 (2004).
12. Doman, T. N. *et al.* Molecular docking and high-throughput screening for novel inhibitors of protein tyrosine phosphatase-1B. *J. Med. Chem.* **45**, 2213–2221 (2002).
13. Warr, W. Report on an NIH Workshop on Ultralarge Chemistry Databases. *ChemRxiv* (2021) doi:10.26434/chemrxiv.14554803.v1.
14. Plowright, A. T. *et al.* Hypothesis driven drug design: Improving quality and effectiveness of the design-make-test-analyse cycle. *Drug Discov. Today* **17**, 56–62 (2012).
15. Schneider, P. *et al.* Rethinking drug design in the artificial intelligence era. *Nat. Rev. Drug Discov.* **19**, 353–364 (2020).
16. Morgan, P. *et al.* Impact of a five-dimensional framework on R&D productivity at AstraZeneca. *Nat. Rev. Drug Discov.* **17**, 167–181 (2018).
17. Zhavoronkov, A. *et al.* Deep learning enables rapid identification of potent DDR1 kinase inhibitors. *Nat. Biotechnol.* **37**, 1038–1040 (2019).
18. Walters, P. W. & Murcko, M. Assessing the impact of generative AI on medicinal chemistry. *Nat. Biotechnol.* **38**, 143–145 (2020).
19. Zhavoronkov, A. & Aspuru-Guzik, A. Reply to 'Assessing the impact of generative AI on medicinal chemistry'. *Nat. Biotechnol.* **38**, 146 (2020).
20. Bender, A. & Cortés-Ciriano, I. Artificial intelligence in drug discovery: what is realistic, what are illusions? Part 1: Ways to make an impact, and why we are not there yet. *Drug Discov. Today* **26**, 511–524 (2021).
21. Chen, H. & Engkvist, O. Has Drug Design Augmented by Artificial Intelligence Become a Reality? *Trends Pharmacol. Sci.* **40**, 806–809 (2019).

22. Danziger, D. J. & Dean, P. M. Automated site-directed drug design: a general algorithm for knowledge acquisition about hydrogen-bonding regions at protein surfaces. *Proc. R. Soc. London. B. Biol. Sci.* **236**, 101–113 (1989).
23. Schneider, G. & Fechner, U. Computer-based de novo design of drug-like molecules. *Nat. Rev. Drug Discov.* **4**, 649–663 (2005).
24. Gillet, V. J., Johnson, A. P., Mata, P. & Sike, S. Automated structure design in 3D. *Tetrahedron Comput. Methodol.* **3**, 681–696 (1990).
25. Nishibata, Y. & Itai, A. Automatic creation of drug candidate structures based on receptor structure. Starting point for artificial lead generation. *Tetrahedron* **47**, 8985–8990 (1991).
26. Pearlman, D. A. & Murcko, M. A. CONCEPTS: New dynamic algorithm forde novo drug suggestion. *J. Comput. Chem.* **14**, 1184–1193 (1993).
27. Douguet, D., Thoreau, E. & Grassy, G. A genetic algorithm for the automated generation of small organic molecules: Drug design using an evolutionary algorithm. *J. Comput. Aided. Mol. Des.* **14**, 449–466 (2000).
28. Schneider, G., Lee, M. L., Stahl, M. & Schneider, P. De novo design of molecular architectures by evolutionary assembly of drug-derived building blocks. *J. Comput. Aided. Mol. Des.* **14**, 487–494 (2000).
29. Segler, M. H. S., Kogej, T., Tyrchan, C. & Waller, M. P. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Cent. Sci.* **4**, 120–131 (2018).
30. Gómez-Bombarelli, R. *et al.* Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Cent. Sci.* **4**, 268–276 (2018).
31. Skinnider, M. A., Stacey, R. G., Wishart, D. S. & Foster, L. J. Chemical language models enable navigation in sparsely populated chemical space. *Nat. Mach. Intell.* **3**, 759–770 (2021).
32. Blum, L. C. & Reymond, J. L. 970 Million druglike small molecules for virtual screening in the chemical universe database GDB-13. *J. Am. Chem. Soc.* **131**, 8732–8733 (2009).
33. Arús-Pous, J. *et al.* Exploring the GDB-13 chemical space using deep generative models. *J. Cheminform.* **11**, 20 (2019).
34. Arús-Pous, J. *et al.* Randomized SMILES strings improve the quality of molecular generative models. *J. Cheminform.* **11**, 71 (2019).
35. Polishchuk, P. CReM: chemically reasonable mutations framework for structure generation. *J. Cheminform.* **12**, 28 (2020).
36. Chan, L., Kumar, R., Verdonk, M. & Poelking, C. 3D pride without 2D prejudice: Bias-controlled multi-level generative models for structure-based ligand design. *arXiv* (2022) doi:10.48550/arxiv.2204.10663.
37. Weininger, D. SMILES, a Chemical Language and Information System: 1: Introduction to Methodology and Encoding Rules. *J. Chem. Inf. Comput. Sci.* **28**, 31–36 (1988).
38. Heller, S. R., McNaught, A., Pletnev, I., Stein, S. & Tchekhovskoi, D. InChI, the IUPAC International Chemical Identifier. *J. Cheminform.* **7**, 23 (2015).
39. O'Boyle, N. & Dalke, A. DeepSMILES: An Adaptation of SMILES for Use in Machine-Learning of Chemical Structures. *ChemRxiv* (2018) doi:10.26434/chemrxiv.7097960.v1.
40. Krenn, M., Häse, F., Nigam, A. K., Friederich, P. & Aspuru-Guzik, A. Self-Referencing Embedded Strings (SELFIES): A 100% robust molecular string representation. *Mach. Learn. Sci. Technol.* **1**, 045024 (2020).
41. Rogers, D. & Hahn, M. Extended-connectivity fingerprints. *J. Chem. Inf. Model.* **50**, 742–754 (2010).
42. Durant, J. L., Leland, B. A., Henry, D. R. & Nourse, J. G. Reoptimization of MDL keys for use in drug discovery. *J. Chem. Inf. Comput. Sci.* **42**, 1273–1280 (2002).
43. Smith, D. H., Carhart, R. E. & Venkataraghavan, R. Atom Pairs as Molecular Features in Structure-Activity Studies: Definition and Applications. *J. Chem. Inf. Comput. Sci.* **25**, 64–73 (1985).
44. Nilakantan, R., Bauman, N., Venkataraghavan, R. & Dixon, J. S. Topological Torsion: A New Molecular

Descriptor for SAR Applications. Comparison with Other Descriptors. *J. Chem. Inf. Comput. Sci.* **27**, 82–85 (1987).

45. Kipf, T. N. & Welling, M. Semi-supervised classification with graph convolutional networks. in *International Conference on Learning Representations* (2017). doi:10.48550/arxiv.1609.02907.
46. Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O. & Dahl, G. E. Neural message passing for quantum chemistry. in *International Conference on Machine Learning* (2017). doi:10.48550/arxiv.1704.01212.
47. Maturana, D. & Scherer, S. VoxNet: A 3D Convolutional Neural Network for real-time object recognition. in *International Conference on Intelligent Robots and Systems* (2015). doi:10.1109/IROS.2015.7353481.
48. Schütt, K. T. *et al.* SchNet: A continuous-filter convolutional neural network for modeling quantum interactions. in *Advances in Neural Information Processing Systems* (2017). doi:10.48550/arxiv.1706.08566.
49. Ruddigkeit, L., Van Deursen, R., Blum, L. C. & Reymond, J. L. Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17. *J. Chem. Inf. Model.* **52**, 2864–2875 (2012).
50. Saldívar-González, F. I., Huerta-García, C. S. & Medina-Franco, J. L. Chemoinformatics-based enumeration of chemical libraries: a tutorial. *J. Cheminform.* **12**, 64 (2020).
51. Rotstein, S. H. & Murcko, M. A. GenStar: A method for de novo drug design. *J. Comput. Aided. Mol. Des.* **7**, 23–43 (1993).
52. Luo, Z., Wang, R. & Lai, L. RASSE: A new method for structure-based drug design. *J. Chem. Inf. Comput. Sci.* **36**, 1187–1194 (1996).
53. Gillet, V., Johnson, A. P., Mata, P., Sike, S. & Williams, P. SPROUT: A program for structure generation. *J. Comput. Aided. Mol. Des.* **7**, 127–153 (1993).
54. Brown, N., Fiscato, M., Segler, M. H. S. & Vaucher, A. C. GuacaMol: Benchmarking Models for de Novo Molecular Design. *J. Chem. Inf. Model.* **59**, 1096–1108 (2019).
55. Gao, W., Fu, T., Sun, J. & Coley, C. W. Sample Efficiency Matters: A Benchmark for Practical Molecular Optimization. *arXiv* (2022) doi:10.48550/arxiv.2206.12411.
56. Jensen, J. H. A graph-based genetic algorithm and generative model/Monte Carlo tree search for the exploration of chemical space. *Chem. Sci.* **10**, 3567–3572 (2019).
57. Steinmann, C. & Jensen, J. H. Using a genetic algorithm to find molecules with good docking scores. *PeerJ Phys. Chem.* **3**, (2021).
58. Nigam, A., Pollice, R. & Aspuru-Guzik, A. Parallel tempered genetic algorithm guided by deep neural networks for inverse molecular design. *Digit. Discov.* **1**, 390–404 (2022).
59. Spiegel, J. O. & Durrant, J. D. AutoGrow4: An open-source genetic algorithm for de novo drug design and lead optimization. *J. Cheminform.* **12**, 25 (2020).
60. Zhou, Z., Kearnes, S., Li, L., Zare, R. N. & Riley, P. Optimization of Molecules via Deep Reinforcement Learning. *Sci. Rep.* **9**, 10752 (2019).
61. Ståhl, N., Falkman, G., Karlsson, A., Mathiason, G. & Boström, J. Deep Reinforcement Learning for Multiparameter Optimization in de novo Drug Design . *J. Chem. Inf. Model.* **59**, 3166–3176 (2019).
62. Horwood, J. & Noutahi, E. Molecular Design in Synthetically Accessible Chemical Space via Deep Reinforcement Learning. *ACS Omega* **5**, 32984–32994 (2020).
63. Fu, T., Gao, W., Coley, C. W. & Sun, J. Reinforced Genetic Algorithm for Structure-based Drug Design. in *Advances in Neural Information Processing Systems* (2022). doi:10.48550/arXiv.2211.16508.
64. Pandey, M. *et al.* The transformational role of GPU computing and deep learning in drug discovery. *Nat. Mach. Intell.* **4**, 211–221 (2022).
65. Krizhevsky, A., Sutskever, I. & Hinton, G. E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **60**, 84–90 (2017).

66. OpenAI. GPT-4 Technical Report. *arXiv* (2023) doi:10.48550/arXiv.2303.08774.
67. Silver, D. *et al.* Mastering the game of Go with deep neural networks and tree search. *Nature* **529**, 484–489 (2016).
68. Chen, H., Engkvist, O., Wang, Y., Olivecrona, M. & Blaschke, T. The rise of deep learning in drug discovery. *Drug Discov. Today* **23**, 1241–1250 (2018).
69. Jumper, J. *et al.* Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583–589 (2021).
70. Chuang, K. V., Gunsalus, L. M. & Keiser, M. J. Learning Molecular Representations for Medicinal Chemistry. *J. Med. Chem.* **63**, 8705–8722 (2020).
71. Elman, J. L. Finding structure in time. *Cogn. Sci.* **14**, 179–211 (1990).
72. Hochreiter, S. & Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **9**, 1735–1780 (1997).
73. Cho, K. *et al.* Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. in *Conference on Empirical Methods in Natural Language Processing* (2014). doi:10.48550/arxiv.1406.1078.
74. Williams, R. J. & Zipser, D. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Comput.* **1**, 270–280 (1989).
75. Amabilino, S., Pogány, P., Pickett, S. D. & Green, D. V. S. Guidelines for Recurrent Neural Network Transfer Learning-Based Molecular Generation of Focused Libraries. *J. Chem. Inf. Model.* **60**, 5699–5713 (2020).
76. Olivecrona, M., Blaschke, T., Engkvist, O. & Chen, H. Molecular de-novo design through deep reinforcement learning. *J. Cheminform.* **9**, 48 (2017).
77. Popova, M., Isayev, O. & Tropsha, A. Deep reinforcement learning for de novo drug design. *Sci. Adv.* **4**, eaap7885 (2018).
78. Xu, M., Ran, T. & Chen, H. De Novo Molecule Design through the Molecular Generative Model Conditioned by 3D Information of Protein Binding Sites. *J. Chem. Inf. Model.* **61**, 3240–3254 (2021).
79. Polykovskiy, D. *et al.* Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models. *Front. Pharmacol.* **11**, 1931 (2020).
80. Ciepliński, T., Danel, T., Podlewska, S. & Jastrzębski, S. Generative Models Should at Least Be Able to Design Molecules That Dock Well: A New Benchmark. *J. Chem. Inf. Model.* **63**, 3238–3247 (2023).
81. Cieplinski, T. smina-docking-benchmark. *GitHub* https://github.com/cieplinski-tobiasz/smina-docking-benchmark.
82. Thomas, M., O'Boyle, N. M., Bender, A. & de Graaf, C. Re-evaluating sample efficiency in de novo molecule generation. *arXiv* (2022) doi:10.48550/arxiv.2212.01385.
83. Martinelli, D. D. Generative machine learning for de novo drug discovery: A systematic review. *Comput. Biol. Med.* **145**, 105403 (2022).
84. Vaswani, A. *et al.* Attention is all you need. in *Advances in Neural Information Processing Systems* (2017). doi:10.48550/arxiv.1706.03762.
85. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. in *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (2018). doi:10.48550/arXiv.1810.04805.
86. Radford, A., Narasimhan, K., Salimans, T. & Sutskever, I. Improving Language Understanding by Generative Pre-Training. *arXiv* (2018).
87. Brown, T. B. *et al.* Language models are few-shot learners. in *Advances in Neural Information Processing Systems* (2020). doi:10.48550/arxiv.2005.14165.
88. He, J. *et al.* Transformer-based molecular optimization beyond matched molecular pairs. *J. Cheminform.* **14**, 18 (2022).

89. Bagal, V., Aggarwal, R., Vinod, P. K. & Priyakumar, U. D. MolGPT: Molecular Generation Using a Transformer-Decoder Model. *J. Chem. Inf. Model.* **62**, 2064–2076 (2021).
90. Grechishnikova, D. Transformer neural network for protein-specific de novo drug generation as a machine translation problem. *Sci. Rep.* **11**, 321 (2021).
91. Sutskever, I., Vinyals, O. & Le, Q. V. Sequence to sequence learning with neural networks. in *Advances in Neural Information Processing Systems* (2014). doi:10.48550/arXiv.1409.3215.
92. Bahdanau, D., Cho, K. H. & Bengio, Y. Neural machine translation by jointly learning to align and translate. in *International Conference on Learning Representations* (2015). doi:10.48550/arXiv.1409.0473.
93. Wu, Y. *et al.* Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *arXiv* (2016) doi:10.48550/arxiv.1609.08144.
94. Ba, J. L., Kiros, J. R. & Hinton, G. E. Layer Normalization. *arXiv* (2016) doi:10.48550/arxiv.1607.06450.
95. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2016). doi:10.1109/CVPR.2016.90.
96. Wang, Y., Zhao, H., Sciabola, S. & Wang, W. cMolGPT: A Conditional Generative Pre-Trained Transformer for Target-Specific De Novo Molecular Generation. *Molecules* **28**, 4430 (2023).
97. Yang, L. *et al.* Transformer-Based Generative Model Accelerating the Development of Novel BRAF Inhibitors. *ACS Omega* **6**, 33864–33873 (2021).
98. Wang, J. *et al.* Multi-constraint molecular generation based on conditional transformer, knowledge distillation and reinforcement learning. *Nat. Mach. Intell.* **3**, 914–922 (2021).
99. Blaschke, T. *et al.* REINVENT 2.0: An AI Tool for de Novo Drug Design. *J. Chem. Inf. Model.* **60**, 5918–5922 (2020).
100. Zheng, S. *et al.* Deep scaffold hopping with multimodal transformer neural networks. *J. Cheminform.* **13**, 87 (2021).
101. Liao, Z., Xie, L., Mamitsuka, H. & Zhu, S. Sc2Mol: a scaffold-based two-step molecule generator with variational autoencoder and transformer. *Bioinformatics* **39**, btac814 (2023).
102. He, J. *et al.* Molecular optimization by capturing chemist's intuition using deep neural networks. *J. Cheminform.* **13**, 26 (2021).
103. Mercado, R. *et al.* Graph networks for molecular design. *Mach. Learn. Sci. Technol.* **2**, 025023 (2021).
104. Zhang, J., Mercado, R., Engkvist, O. & Chen, H. Comparative Study of Deep Generative Models on Chemical Space Coverage. *J. Chem. Inf. Model.* **61**, 2572–2581 (2021).
105. Elton, D. C., Boukouvalas, Z., Fuge, M. D. & Chung, P. W. Deep learning for molecular design—a review of the state of the art. *Mol. Syst. Des. Eng.* **4**, 828–849 (2019).
106. Jin, W., Barzilay, R. & Jaakkola, T. Junction tree variational autoencoder for molecular graph generation. in *International Conference on Machine Learning* (2018). doi:10.48550/arXiv.1802.04364.
107. Maziarz, K. *et al.* Learning to Extend Molecular Scaffolds with Structural Motifs. in *International Conference on Learning Representations* (2021). doi:10.48550/arXiv.2103.03864.
108. Arjovsky, M., Chintala, S. & Bottou, L. Wasserstein generative adversarial networks. in *International Conference on Machine Learning* (2017). doi:10.48550/arXiv.1701.07875.
109. Guimaraes, G. L., Sanchez-Lengeling, B., Outeiral, C., Farias, P. L. C. & Aspuru-Guzik, A. Objective-Reinforced Generative Adversarial Networks (ORGAN) for Sequence Generation Models. *arXiv* (2017) doi:10.48550/arXiv.1705.10843.
110. De Cao, N. & Kipf, T. MolGAN: An implicit generative model for small molecular graphs. *arXiv* (2018) doi:10.48550/arXiv.1805.11973.
111. Méndez-Lucio, O., Baillif, B., Clevert, D. A., Rouquié, D. & Wichard, J. De novo generation of hit-like molecules from gene expression signatures using artificial intelligence. *Nat. Commun.* **11**, 1–10 (2020).

112. Skalic, M., Sabbadin, D., Sattarov, B., Sciabola, S. & De Fabritiis, G. From Target to Drug: Generative Modeling for the Multimodal Structure-Based Ligand Design. *Mol. Pharm.* **16**, 4282–4291 (2019).
113. Kotsias, P. C. *et al.* Direct steering of de novo molecular generation with descriptor conditional recurrent neural networks. *Nat. Mach. Intell.* **2**, 254–265 (2020).
114. Yang, X., Zhang, J., Yoshizoe, K., Terayama, K. & Tsuda, K. ChemTS: an efficient python library for de novo molecular generation. *Sci. Technol. Adv. Mater.* **18**, 972 (2017).
115. Xu, Z., Wauchope, O. R. & Frank, A. T. Navigating Chemical Space by Interfacing Generative Artificial Intelligence and Molecular Docking. *J. Chem. Inf. Model.* **61**, 5589–5600 (2021).
116. Ahn, S., Kim, J., Lee, H. & Shin, J. Guiding Deep Molecular Optimization with Genetic Exploration. in *Advances in Neural Information Processing Systems* (2020). doi:10.48550/arXiv.2007.04897.
117. Winter, R. *et al.* Efficient multi-objective molecular optimization in a continuous latent space. *Chem. Sci.* **10**, 8016–8024 (2019).
118. Griffiths, R. R. & Hernández-Lobato, J. M. Constrained Bayesian optimization for automatic chemical design using variational autoencoders. *Chem. Sci.* **11**, 577–586 (2020).
119. Kusner, M. J., Paige, B., Miguel Hernández-Lobato, J. & Hernández-Lobato, J. M. Grammar Variational Autoencoder. in *International conference on machine learning* (2017). doi:10.48550/arXiv.1703.01925.
120. Sutton, R. S. & Barto, A. G. *Reinforcement Learning: an introduction*. (MIT Press, 2018).
121. Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.* **8**, 229–256 (1992).
122. Neil, D. *et al.* Exploring Deep Recurrent Models with Reinforcement Learning for Molecule Design. in *International Conference on Learning Representations* (2018).
123. Atance, S. R., Diez, J. V., Engkvist, O., Olsson, S. & Mercado, R. De Novo Drug Design Using Reinforcement Learning with Graph-Based Deep Generative Models. *J. Chem. Inf. Model.* **62**, 4863–4872 (2022).
124. Huang, K. *et al.* Therapeutics Data Commons: Machine Learning Datasets and Tasks for Drug Discovery and Development. in *Advances in Neural Information Processing Systems* (2021). doi:10.48550/arXiv.2102.09548.
125. Jaques, N. *et al.* Sequence Tutor: Conservative Fine-Tuning of Sequence Generation Models with KL-control. in *International Conference on Machine Learning* (2017). doi:10.48550/arXiv.1611.02796.
126. Boitreaud, J., Mallet, V., Oliver, C. & Waldispuhl, J. OptiMol: Optimization of binding affinities in chemical space for drug discovery. *J. Chem. Inf. Model.* **60**, 5658–5666 (2020).
127. Sanchez-Lengeling, B., Outeiral, C., Guimaraes, G. L. & Aspuru-Guzik, A. Optimizing distributions over molecular space. An Objective-Reinforced Generative Adversarial Network for Inverse-design Chemistry (ORGANIC). *ChemRxiv* (2017) doi:10.26434/chemrxiv.5309668.v3.
128. Korshunova, M. *et al.* Generative and reinforcement learning approaches for the automated de novo design of bioactive compounds. *Commun. Chem.* **5**, 129 (2022).
129. Verlinde, C. L. & Hol, W. G. Structure-based drug design: progress, results and challenges. *Structure* **2**, 577–587 (1994).
130. Ripphausen, P., Stumpfe, D. & Bajorath, J. Analysis of structure-based virtual screening studies and characterization of identified active compounds. *Future Med. Chem.* **4**, 603–613 (2012).
131. Ballante, F., Kooistra, A. J., Kampen, S., de Graaf, C. & Carlsson, J. Structure-Based Virtual Screening for Ligands of G Protein–Coupled Receptors: What Can Molecular Docking Do for You? *Pharmacol. Rev.* **73**, 1698–1736 (2021).
132. Congreve, M., Murray, C. W. & Blundell, T. L. Keynote review: Structural biology and drug discovery. *Drug Discov. Today* **10**, 895–907 (2005).

133. Sheridan, R. P. *et al.* Experimental Error, Kurtosis, Activity Cliffs, and Methodology: What Limits the Predictivity of Quantitative Structure-Activity Relationship Models? *J. Chem. Inf. Model.* **60**, 1969–1982 (2020).
134. Renz, P., Van Rompaey, D., Wegner, J. K., Hochreiter, S. & Klambauer, G. On failure modes in molecule generation and optimization. *Drug Discov. Today Technol.* **32–33**, 55–63 (2020).
135. Langevin, M., Vuilleumier, R. & Bianciotto, M. Explaining and avoiding failure modes in goal-directed generation of small molecules. *J. Cheminform.* **14**, 20 (2022).
136. Congreve, M., de Graaf, C., Swain, N. A. & Tate, C. G. Impact of GPCR Structures on Drug Discovery. *Cell* **181**, 81–91 (2020).
137. Berman, H. M. *et al.* The Protein Data Bank. *Nucleic Acids Res.* **28**, 235–242 (2000).
138. Baek, M. *et al.* Accurate prediction of protein structures and interactions using a three-track neural network. *Science* **373**, 871–876 (2021).
139. Chowdhury, R. *et al.* Single-sequence protein structure prediction using a language model and deep learning. *Nat. Biotechnol.* **40**, 1617–1623 (2022).
140. Hekkelman, M. L., Vries, I. de, Joosten, R. P. & Perrakis, A. AlphaFill: enriching the AlphaFold models with ligands and co-factors. *Nat. Methods* **20**, 205–213 (2023).
141. Su, M. *et al.* Comparative Assessment of Scoring Functions: The CASF-2016 Update. *J. Chem. Inf. Model.* **59**, 895–913 (2019).
142. Enyedy, I. J. & Egan, W. J. Can we use docking and scoring for hit-to-lead optimization? *J. Comput. Aided. Mol. Des.* **22**, 161–168 (2008).
143. Wang, R., Fang, X., Lu, Y. & Wang, S. The PDBbind database: Collection of binding affinities for protein-ligand complexes with known three-dimensional structures. *J. Med. Chem.* **47**, 2977–2980 (2004).
144. Henzler-Wildman, K. & Kern, D. Dynamic personalities of proteins. *Nature* **450**, 964–972 (2007).
145. Sterling, T. & Irwin, J. J. ZINC 15 - Ligand Discovery for Everyone. *J. Chem. Inf. Model.* **55**, 2324–2337 (2015).
146. Jeon, W. & Kim, D. Autonomous molecule generation using reinforcement learning and docking to develop potential novel inhibitors. *Sci. Rep.* **10**, 22104 (2020).
147. Thomas, M., Smith, R. T., O'Boyle, N. M., de Graaf, C. & Bender, A. Comparison of structure- and ligand-based scoring functions for deep generative models: a GPCR case study. *J. Cheminform.* **13**, 39 (2021).
148. Guo, J. *et al.* DockStream: a docking wrapper to enhance de novo molecular design. *J. Cheminform.* **13**, 89 (2021).
149. García-Ortegón, M. *et al.* DOCKSTRING: Easy Molecular Docking Yields Better Benchmarks for Ligand Design. *J. Chem. Inf. Model.* **62**, 3486–3502 (2022).
150. Thomas, M. MolScore. *GitHub* https://github.com/MorganCThomas/MolScore/ (2021).
151. Ragoza, M., Masuda, T. & Koes, D. R. Generating 3D molecules conditional on receptor binding sites with deep generative models. *Chem. Sci.* **13**, 2701–2713 (2022).
152. Wang, L. *et al.* A pocket-based 3D molecule generative model fueled by experimental electron density. *Sci. Rep.* **12**, 15100 (2022).
153. Drotár, P., Jamasb, A. R., Day, B., Atˇ Alina Cangea, C. & Liò, P. Structure-aware generation of drug-like molecules. in *Advances in Neural Information Processing Systems* (2021). doi:10.48550/arXiv.2111.04107.
154. Luo, S., Guan, J., Ma, J. & Peng, J. A 3D Generative Model for Structure-Based Drug Design. in *Advances in Neural Information Processing Systems* (2021). doi:10.48550/arXiv.2203.10446.
155. Powers, A. S., Yu, H. H., Suriana, P. & Dror, R. O. Fragment-Based Ligand Generation Guided By Geometric Deep Learning On Protein-Ligand Structure. *bioRxiv* (2022) doi:10.1101/2022.03.17.484653.

156. Peng, X. *et al.* Pocket2Mol: Efficient Molecular Sampling Based on 3D Protein Pockets. in *International Conference on Machine* (2022). doi:10.48550/arxiv.2205.07249.
157. Molteni, V. *et al.* Identification of a small-molecule binding site at the dimer interface of the HIV integrase catalytic domain. *Acta Crystallogr. Sect. D Biol. Crystallogr.* **57**, 536–544 (2001).
158. Cheong, C. G. *et al.* Crystal Structures of Human Bifunctional Enzyme Aminoimidazole-4-carboxamide Ribonucleotide Transformylase/IMP Cyclohydrolase in Complex with Potent Sulfonyl-containing Antifolates. *J. Biol. Chem.* **279**, 18034–18045 (2004).
159. Francoeur, P. G. *et al.* Three-dimensional convolutional neural networks and a crossdocked data set for structure-based drug design. *J. Chem. Inf. Model.* **60**, 4200–4215 (2020).
160. Schneuing, A. *et al.* Structure-based Drug Design with Equivariant Diffusion Models. *arXiv* (2022) doi:10.48550/arxiv.2210.13695.
161. Li, X., Xu, Y., Yao, H. & Lin, K. Chemical space exploration based on recurrent neural networks: applications in discovering kinase inhibitors. *J. Cheminform.* **12**, 42 (2020).
162. Benhenda, M. ChemGAN challenge for drug discovery: can AI reproduce natural chemical diversity? *arXiv* (2017) doi:10.48550/arXiv.1708.08227.
163. Gobbi, A. & Lee, M. L. DISE: Directed sphere exclusion. *J. Chem. Inf. Comput. Sci.* **43**, 317–323 (2003).
164. Baell, J. B. & Holloway, G. A. rd_filters. *GitHub* doi:10.1021/JM901137J.
165. Preuer, K., Renz, P., Unterthiner, T., Hochreiter, S. & Klambauer, G. Fréchet ChemNet Distance: A Metric for Generative Models for Molecules in Drug Discovery. *J. Chem. Inf. Model.* **58**, 1736–1741 (2018).
166. Mayr, A. *et al.* Large-scale comparison of machine learning methods for drug target prediction on ChEMBL. *Chem. Sci.* **9**, 5441–5451 (2018).
167. Blaschke, T., Engkvist, O., Bajorath, J. & Chen, H. Memory-assisted reinforcement learning for diverse molecular de novo design. *J. Cheminform.* **12**, 68 (2020).
168. Ertl, P. An algorithm to identify functional groups in organic molecules. *J. Cheminform.* **9**, 36 (2017).
169. Degen, J., Wegscheid-Gerlach, C., Zaliani, A. & Rarey, M. On the Art of Compiling and Using 'Drug-Like' Chemical Fragment Spaces. *ChemMedChem* **3**, 1503–1507 (2008).
170. Bemis, G. W. & Murcko, M. A. The properties of known drugs. 1. Molecular frameworks. *J. Med. Chem.* **39**, 2887–2893 (1996).
171. Wildman, S. A. & Crippen, G. M. Prediction of Physicochemical Parameters by Atomic Contributions. *J. Chem. Inf. Comput. Sci.* **39**, 868–873 (1999).
172. Ertl, P. & Schuffenhauer, A. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *J. Cheminform.* **1**, 8 (2009).
173. Bickerton, G. R., Paolini, G. V., Besnard, J., Muresan, S. & Hopkins, A. L. Quantifying the chemical beauty of drugs. *Nat. Chem.* **4**, 90–98 (2012).
174. Dixon, S. L. & Koehler, R. T. The hidden component of size in two-dimensional fragment descriptors: Side effects on sampling in bioactive libraries. *J. Med. Chem.* **42**, 2887–2900 (1999).
175. Grisoni, F. *et al.* Combining generative artificial intelligence and on-chip synthesis for de novo drug design. *Sci. Adv.* **7**, eabg3338 (2021).
176. Yoshikawa, N. *et al.* Population-based De Novo Molecule Generation, Using Grammatical Evolution. *Chem. Lett.* **47**, 1431–1434 (2018).
177. Jiang, D. *et al.* Could graph neural networks learn better molecular representation for drug discovery? A comparison study of descriptor-based and graph-based models. *J. Cheminform.* **13**, 12 (2021).
178. Mysinger, M. M., Carchia, M., Irwin, J. J. & Shoichet, B. K. Directory of useful decoys, enhanced (DUD-E): Better ligands and decoys for better benchmarking. *J. Med. Chem.* **55**, 6582–6594 (2012).
179. Thakkar, A. *et al.* Retrosynthetic accessibility score (RAscore) – rapid machine learned synthesizability

classification from AI driven retrosynthetic planning. *Chem. Sci.* **12**, 3339–3349 (2021).

180. Coley, C. W., Rogers, L., Green, W. H. & Jensen, K. F. SCScore: Synthetic Complexity Learned from a Reaction Corpus. *J. Chem. Inf. Model.* **58**, 252–261 (2018).
181. Doak, B. C., Over, B., Giordanetto, F. & Kihlberg, J. Oral druggable space beyond the rule of 5: Insights from drugs and clinical candidates. *Chem. Biol.* **21**, 1115–1142 (2014).
182. DeGoey, D. A., Chen, H.-J., Cox, P. B. & Wendt, M. D. Beyond the Rule of 5: Lessons Learned from AbbVie's Drugs and Compound Collection: Miniperspective. *J. Med. Chem.* **61**, 2636–2651 (2018).
183. Zimmermann, M., Zimmermann-Kogadeeva, M., Wegmann, R. & Goodman, A. L. Separating host and microbiome contributions to drug pharmacokinetics and toxicity. *Science* **363**, eaat9931 (2019).
184. Sheridan, R. P., Feuston, B. P., Maiorov, V. N. & Kearsley, S. K. Similarity to molecules in the training set is a good discriminator for prediction accuracy in QSAR. *J. Chem. Inf. Comput. Sci.* **44**, 1912–1928 (2004).
185. Miljković, F. & Bajorath, J. Computational Analysis of Kinase Inhibitors Identifies Promiscuity Cliffs across the Human Kinome. *ACS Omega* **3**, 17295–17308 (2018).
186. Richard, A. M. *et al.* The Tox21 10K Compound Library: Collaborative Chemistry Advancing Toxicology. *Chem. Res. Toxicol.* **34**, 189–216 (2021).
187. Valdes, G. & Interian, Y. Comment on 'Deep convolutional neural network with transfer learning for rectum toxicity prediction in cervical cancer radiotherapy: A feasibility study'. *Phys. Med. Biol.* **63**, 068001 (2018).
188. Smith, G. F. Designing Drugs to Avoid Toxicity. in *Progress in Medicinal Chemistry* vol. 50 1–47 (Elsevier B.V., 2011).
189. Manevski, N., King, L., Pitt, W. R., Lecomte, F. & Toselli, F. Metabolism by Aldehyde Oxidase: Drug Design and Complementary Approaches to Challenges in Drug Discovery. *J. Med. Chem.* **62**, 10955–10994 (2019).
190. Zhang, J. W., Xiao, W., Gao, Z. T., Yu, Z. T. & Zhang, J. Y. Metabolism of c-Met kinase inhibitors containing quinoline by aldehyde oxidase, electron donating, and steric hindrance effect. *Drug Metab. Dispos.* **46**, 1847–1855 (2018).
191. Marin Zapata, P. A. *et al.* Cell morphology-guided de novo hit design by conditioning GANs on phenotypic image features. *Digit. Discov.* **2**, 91–102 (2023).
192. Goodhart, C. A. E. Problems of Monetary Management: The UK Experience. in *Monetary Theory and Practice* vol. 1 91–121 (1984).
193. Stewart, K. D., Shiroda, M. & James, C. A. Drug Guru: A computer software program for drug design using medicinal chemistry rules. *Bioorganic Med. Chem.* **14**, 7011–7022 (2006).
194. Merk, D., Grisoni, F., Friedrich, L. & Schneider, G. Tuning artificial intelligence on the de novo design of natural-product-inspired retinoid X receptor modulators. *Commun. Chem.* **1**, 68 (2018).
195. Yang, Y. *et al.* Discovery of Highly Potent, Selective, and Orally Efficacious p300/CBP Histone Acetyltransferases Inhibitors. *J. Med. Chem.* **63**, 1337–1360 (2020).
196. Hua, Y. *et al.* Effective Reaction-Based De Novo Strategy for Kinase Targets: A Case Study on MERTK Inhibitors. *J. Chem. Inf. Model.* **62**, 1654–1668 (2022).
197. Moret, M. *et al.* Leveraging molecular structure and bioactivity with chemical language models for de novo drug design. *Nat. Commun.* **14**, 114 (2023).
198. Goldman, B., Kearnes, S., Kramer, T., Riley, P. & Walters, W. P. Defining Levels of Automated Chemical Design. *J. Med. Chem.* **65**, 7073–7087 (2022).
199. RDKit. Open-source cheminformatics. http://www.rdkit.org.
200. Paszke, A. *et al.* PyTorch: An Imperative Style, High-Performance Deep Learning Library. in *Advances in Neural Information Processing Systems* (2019). doi:10.48550/arXiv.1912.01703.
201. Genheden, S. *et al.* AiZynthFinder: a fast, robust and flexible open-source software for retrosynthetic

planning. *J. Cheminform.* **12**, 70 (2020).
202. Walters, W. P. & Barzilay, R. Applications of Deep Learning in Molecule Generation and Molecular Property Prediction. *Acc. Chem. Res.* **54**, 263–270 (2021).
203. Pedregosa, F. *et al.* Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
204. Mendez, D. *et al.* ChEMBL: Towards direct deposition of bioassay data. *Nucleic Acids Res.* **47**, D930–D940 (2019).
205. Thomas, M., Abrudan, A., Hosseini-Gerami, L. & Bender, A. PIDGINv5. *Zenodo* https://zenodo.org/record/7547691 (2023) doi:10.5281/zenodo.7547691.
206. Dask Development Team. Dask: Library for dynamic task scheduling. https://dask.org (2016).
207. Grant, J. A., Gallardo, M. A. & Pickup, B. T. A fast method of molecular shape comparison: A simple application of a Gaussian description of molecular shape. *J. Comput. Chem.* **17**, 1653–1666 (1996).
208. Tosco, P., Balle, T. & Shiri, F. Open3DALIGN: An open-source software aimed at unsupervised ligand alignment. *J. Comput. Aided. Mol. Des.* **25**, 777–783 (2011).
209. Langevin, M. *et al.* Impact of Applicability Domains to Generative Artificial Intelligence. *ACS Omega* **8**, 23148–23167 (2023).
210. Mervin, L. H. *et al.* Target prediction utilising negative bioactivity data covering large chemical space. *J. Cheminform.* **7**, 51 (2015).
211. Yang, K. *et al.* Analyzing Learned Molecular Representations for Property Prediction. *J. Chem. Inf. Model.* **59**, 3370–3388 (2019).
212. Friesner, R. A. *et al.* Glide: A New Approach for Rapid, Accurate Docking and Scoring. 1. Method and Assessment of Docking Accuracy. *J. Med. Chem.* **47**, 1739–1749 (2004).
213. Korb, O., Stützle, T. & Exner, T. E. An ant colony optimization approach to flexible protein–ligand docking. *Swarm Intell.* **1**, 115–134 (2007).
214. Jones, G., Willett, P., Glen, R. C., Leach, A. R. & Taylor, R. Development and validation of a genetic algorithm for flexible docking. *J. Mol. Biol.* **267**, 727–748 (1997).
215. McGann, M. R., Almond, H. R., Nicholls, A., Grant, J. A. & Brown, F. K. Gaussian docking functions. *Biopolymers* **68**, 76–90 (2003).
216. Koes, D. R., Baumgartner, M. P. & Camacho, C. J. Lessons learned in empirical scoring with smina from the CSAR 2011 benchmarking exercise. *J. Chem. Inf. Model.* **53**, 1893–1904 (2013).
217. Ropp, P. J. *et al.* Gypsum-DL: An open-source program for preparing small-molecule libraries for structure-based virtual screening. *J. Cheminform.* **11**, 34 (2019).
218. Schrödinger Release 2019-4. LigPrep.
219. Shelley, J. C. *et al.* Epik: A software program for pKa prediction and protonation state generation for drug-like molecules. *J. Comput. Aided. Mol. Des.* **21**, 681–691 (2007).
220. Milletti, F., Storchi, L., Sforna, G. & Cruciani, G. New and original pKa prediction method using grid molecular interaction fields. *J. Chem. Inf. Model.* **47**, 2172–2181 (2007).
221. Liu, X. *et al.* DrugEx v2: de novo design of drug molecules by Pareto-based multi-objective reinforcement learning in polypharmacology. *J. Cheminform.* **13**, 85 (2021).
222. Bento, A. P. *et al.* An open source chemical structure curation pipeline using RDKit. *J. Cheminform.* **12**, 51 (2020).
223. Bertz, S. H. The First General Index of Molecular Complexity. *J. Am. Chem. Soc.* **103**, 3599–3601 (1981).
224. Hawkins, P. C. D., Skillman, A. G. & Nicholls, A. Comparison of shape-matching and docking as virtual screening tools. *J. Med. Chem.* **50**, 74–82 (2007).
225. 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. *J. Chem. Inf. Model.* **50**, 572–584 (2010).
226. Jung, S., Vatheuer, H. & Czodrowski, P. VSFlow: an open-source ligand-based virtual screening tool. *J. Cheminform.* **15**, 40 (2023).
227. Ropp, P. J., Kaminsky, J. C., Yablonski, S. & Durrant, J. D. Dimorphite-DL: An open-source program for enumerating the ionization states of drug-like small molecules. *J. Cheminform.* **11**, 14 (2019).
228. mn-am. CORINA classic.
229. McGann, M. FRED pose prediction and virtual screening accuracy. *J. Chem. Inf. Model.* **51**, 578–596 (2011).
230. McGann, M. FRED and HYBRID docking performance on standardized datasets. *J. Comput. Aided. Mol. Des.* **26**, 897–906 (2012).
231. Chen, T. & Guestrin, C. XGBoost: A scalable tree boosting system. in *International Conference on Knowledge Discovery and Data Mining* (2016). doi:10.1145/2939672.2939785.
232. Segler, M. H. S., Preuss, M. & Waller, M. P. Planning chemical syntheses with deep neural networks and symbolic AI. *Nature* **555**, 604–610 (2018).
233. Thomas, M. *et al.* Applications of Artificial Intelligence in Drug Design: Opportunities and Challenges. *Methods Mol. Biol.* **2390**, 1–59 (2022).
234. White, A. D. molbloom: quick assessment of compound purchasability with bloom filters. (2022) doi:10.5281/zenodo.7426402.
235. Irwin, J. J. *et al.* ZINC20 - A Free Ultralarge-Scale Chemical Database for Ligand Discovery. *J. Chem. Inf. Model.* **60**, 6065–6073 (2020).
236. Walters, P. Silly_walks: Identifying silly molecules. *GitHub* https://github.com/PatWalters/silly_walks.
237. Virtanen, P. *et al.* SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods* **17**, 261–272 (2020).
238. Schrödinger. The PyMOL Molecular Graphics System.
239. CHEMBL database release 28. http://ftp.ebi.ac.uk/pub/databases/chembl/ChEMBLdb/releases/chembl_28 (2021) doi:10.6019/CHEMBL.database.28.
240. Diversity Libraries - Enamine. https://enamine.net/hit-finding/diversity-libraries.
241. Targeted Libraries - Enamine. https://enamine.net/hit-finding/focused-libraries.
242. Sanguinetti, M. C. & Tristani-Firouzi, M. hERG potassium channels and cardiac arrhythmia. *Nature* **440**, 463–469 (2006).
243. Li, P. *et al.* Discovery of a tetracyclic quinoxaline derivative as a potent and orally active multifunctional drug candidate for the treatment of neuropsychiatric and neurological disorders. *J. Med. Chem.* **57**, 2670–2682 (2014).
244. 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. *J. Cheminform.* **14**, 68 (2022).
245. Pajouhesh, H. & Lenz, G. R. Medicinal chemical properties of successful central nervous system drugs. *NeuroRx* **2**, 541–553 (2005).
246. Casey, A. B., Cui, M., Booth, R. G. & Canal, C. E. "Selective" serotonin 5-HT2A receptor antagonists. *Biochem. Pharmacol.* **200**, 115028 (2022).
247. Blair, D. T. & Dauner, A. Extrapyramidal symptoms are serious side-effects of antipsychotic and other drugs. *Nurse Pract.* **17**, 56–67 (1992).
248. Kooistra, A. . & de Graaf, C. GPCR Tree Mapper (in preparation).
249. Thomas, M., Bender, A. & de Graaf, C. Integrating structure-based approaches in generative molecular design. *Curr. Opin. Struct. Biol.* **79**, 102559 (2023).

250. Guo, J. *et al.* Improving de novo molecular design with curriculum learning. *Nat. Mach. Intell.* **4**, 555–563 (2022).
251. Cleves, A. E. & Jain, A. N. Effects of inductive bias on computational evaluations of ligand-based modeling and on drug discovery. *J. Comput. Aided. Mol. Des.* **22**, 147–159 (2008).
252. Wallach, I. & Heifets, A. Most Ligand-Based Classification Benchmarks Reward Memorization Rather than Generalization. *J. Chem. Inf. Model.* **58**, 916–932 (2018).
253. Kitchen, D. B., Decornez, H., Furr, J. R. & Bajorath, J. Docking and scoring in virtual screening for drug discovery: Methods and applications. *Nat. Rev. Drug Discov.* **3**, 935–949 (2004).
254. 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 (2009).
255. Bordogna, A., Pandini, A. & Bonati, L. Predicting the accuracy of protein-ligand docking on homology models. *J. Comput. Chem.* **32**, 81–98 (2011).
256. Du, H., Brender, J. R., Zhang, J. & Zhang, Y. Protein structure prediction provides comparable performance to crystallographic structures in docking-based virtual screening. *Methods* **71**, 77–84 (2015).
257. Callaway, E. 'It will change everything': DeepMind's AI makes gigantic leap in solving protein structures. *Nature* **588**, 203–204 (2020).
258. Jin, W., Yang, K., Barzilay, R. & Jaakkola, T. Learning Multimodal Graph-to-Graph Translation for Molecular Optimization. *arXiv* (2018) doi:10.48550/arXiv.1812.01070.
259. Wang, S. *et al.* Structure of the D2 dopamine receptor bound to the atypical antipsychotic drug risperidone. *Nature* **555**, 269–273 (2018).
260. Ghanakota, P. *et al.* 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 (2020).
261. Dixon, S. L. *et al.* AutoQSAR: An automated machine learning tool for best-practice quantitative structure-activity relationship modeling. *Future Med. Chem.* **8**, 1825–1839 (2016).
262. Sun, J. *et al.* ExCAPE-DB: an integrated large scale dataset facilitating Big Data analysis in chemogenomics. *J. Cheminform.* **9**, 17 (2017).
263. Wang, R., Fu, Y. & Lai, L. A new atom-additive method for calculating partition coefficients. *J. Chem. Inf. Comput. Sci.* **37**, 615–621 (1997).
264. Kalgutkar, A. S. & Soglia, J. R. Minimising the potential for metabolic activation in drug discovery. *Expert Opin. Drug Metab. Toxicol.* **1**, 91–142 (2005).
265. Kalgutkar, A. *et al.* A Comprehensive Listing of Bioactivation Pathways of Organic Functional Groups. *Curr. Drug Metab.* **6**, 161–225 (2005).
266. O'Boyle, N. M. No charge - A simple approach to neutralising charged molecules. *Noel O'Blog* https://baoilleach.blogspot.com/2019/12/no-charge-simple-approach-to.html (2019).
267. Kooistra, A. J. *et al.* Function-specific virtual screening for GPCR ligands using a combined scoring method. *Sci. Rep.* **6**, 28288 (2016).
268. Vass, M. *et al.* Molecular interaction fingerprint approaches for GPCR drug discovery. *Curr. Opin. Pharmacol.* **30**, 59–68 (2016).
269. Kim, S. *et al.* PubChem 2019 update: improved access to chemical data. *Nucleic Acids Res.* **47**, (2019).
270. Kingma, D. P. & Ba, J. L. Adam: A method for stochastic optimization. in *International Conference on Learning Representations* (2015). doi:10.48550/arXiv.1412.6980.
271. Madhavi Sastry, G., Adzhigirey, M., Day, T., Annabhimoju, R. & Sherman, W. Protein and ligand preparation: Parameters, protocols, and influence on virtual screening enrichments. *J. Comput. Aided. Mol. Des.* **27**, 221–234 (2013).

272. Sondergaard, C. R., Olsson, M. H. M., Rostkowski, M. & Jensen, J. H. Improved treatment of ligands and coupling effects in empirical calculation and rationalization of p K a values. *J. Chem. Theory Comput.* **7**, 2284–2295 (2011).
273. Roos, K. *et al.* OPLS3e: Extending Force Field Coverage for Drug-Like Small Molecules. *J. Chem. Theory Comput.* **15**, 1863–1874 (2019).
274. Bender, A. How to Lie With Computational Predictive Models in Drug Discovery. *DrugDiscovery.NET - AI in Drug Discovery* http://www.drugdiscovery.net/2020/10/13/how-to-lie-with-computational-predictive-models-in-drug-discovery/ (2020).
275. Sayle, R. A. 2d similarity, diversity and clustering in rdkit. in *RDKit UGM* (2019).
276. McInnes, L., Healy, J. & Melville, J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv* (2018) doi:10.48550/arXiv.1802.03426.
277. Sauer, W. H. B. & Schwarz, M. K. Molecular shape diversity of combinatorial libraries: A prerequisite for broad bioactivity. *J. Chem. Inf. Comput. Sci.* **43**, 987–1003 (2003).
278. Riniker, S. & Landrum, G. A. Better Informed Distance Geometry: Using What We Know to Improve Conformation Generation. *J. Chem. Inf. Model.* **55**, 2562–2574 (2015).
279. Deng, Z., Chuaqui, C. & Singh, J. Structural Interaction Fingerprint (SIFt): A Novel Method for Analyzing Three-Dimensional Protein-Ligand Binding Interactions. *J. Med. Chem.* **47**, 337–344 (2004).
280. Xiao, J. *et al.* Discovery, optimization, and characterization of novel D2 dopamine receptor selective antagonists. *J. Med. Chem.* **57**, 3450–3463 (2014).
281. Tomberg, A. & Boström, J. Can easy chemistry produce complex, diverse, and novel molecules? *Drug Discov. Today* **25**, 2174–2181 (2020).
282. Kramer, C., Fuchs, J. E. & Liedl, K. R. Strong nonadditivity as a key structure-activity relationship feature: Distinguishing structural changes from assay artifacts. *J. Chem. Inf. Model.* **55**, 483–494 (2015).
283. Gao, W. & Coley, C. W. The Synthesizability of Molecules Proposed by Generative Models. *J. Chem. Inf. Model.* **60**, 5714–5723 (2020).
284. Danel, T. *et al.* De Novo Drug Design with a Docking Score Proxy. in *Advances in Neural Information Processing Systems - Machine learning for molecules workshop* (2020).
285. Pan, Y., Huang, N., Cho, S. & MacKerell, A. D. Consideration of molecular weight during compound selection in virtual target-based database screening. *J. Chem. Inf. Comput. Sci.* **43**, 267–272 (2003).
286. Carta, G., Knox, A. J. S. & Lloyd, D. G. Unbiasing scoring functions: A new normalization and rescoring strategy. *J. Chem. Inf. Model.* **47**, 1564–1571 (2007).
287. Kaczor, A. A. *et al.* Structure-Based Virtual Screening for Dopamine D2 Receptor Ligands as Potential Antipsychotics. *ChemMedChem* **11**, 718–729 (2016).
288. Moret, M., Helmstädter, M., Grisoni, F., Schneider, G. & Merk, D. Beam Search for Automated Design and Scoring of Novel ROR Ligands with Machine Intelligence. *Angew. Chemie - Int. Ed.* **60**, 19477–19482 (2021).
289. Flam-Shepherd, D., Zhu, K. & Aspuru-Guzik, A. Language models can learn complex molecular distributions. *Nat. Commun.* **13**, 3293 (2022).
290. Sutton, R. S. & Barto, A. G. Policy Gradient Methods. in *Reinforcement Learning: An Introduction* 326 (MIT Press, 2018).
291. Nigam, A., Pollice, R., Krenn, M., Gomes, G. D. P. & Aspuru-Guzik, A. Beyond generative models: Superfast traversal, optimization, novelty, exploration and discovery (STONED) algorithm for molecules using SELFIES. *Chem. Sci.* **12**, 7079–7090 (2021).
292. Ma, B. *et al.* Structure-Based de Novo Molecular Generator Combined with Artificial Intelligence and Docking Simulations. *J. Chem. Inf. Model.* **61**, 3304–3313 (2021).

293. Krishnan, S. R., Bung, N., Bulusu, G. & Roy, A. Accelerating de Novo Drug Design against Novel Proteins Using Deep Learning. *J. Chem. Inf. Model.* **61**, 621–630 (2021).
294. Fialková, V. *et al.* LibINVENT: Reaction-based Generative Scaffold Decoration for in Silico Library Design. *J. Chem. Inf. Model.* **62**, 2046–2063 (2022).
295. Moret, M., Friedrich, L., Grisoni, F., Merk, D. & Schneider, G. Generating customized compound libraries for drug discovery with machine intelligence. *ChemRxiv* (2019) doi:10.26434/CHEMRXIV.10119299.V1.
296. BenevolentAI. GuacaMol Baselines. *GitHub* https://github.com/BenevolentAI/guacamol_baselines.
297. Parisotto, E. *et al.* Stabilizing transformers for reinforcement learning. in *International Conference on Machine Learning* (2020). doi:10.48550/arxiv.1910.06764.
298. Dai, Z. *et al.* Transformer-XL: Attentive language models beyond a fixed-length context. *arXiv* (2020) doi:10.48550/arXiv.1901.02860.
299. Manglik, A. *et al.* Crystal structure of the μ-opioid receptor bound to a morphinan antagonist. *Nature* **485**, 321–326 (2012).
300. Zhang, H. *et al.* Structure of the angiotensin receptor revealed by serial femtosecond crystallography. *Cell* **161**, 833–844 (2015).
301. Rappas, M. *et al.* Comparison of Orexin 1 and Orexin 2 Ligand Binding Modes Using X-ray Crystallography and Computational Analysis. *J. Med. Chem.* **63**, 1528–1543 (2020).
302. Schrödinger Release 2019-4. Protein Preparation Wizard.
303. Ashton, M. *et al.* Identification of Diverse Database Subsets using Property-Based and Fragment-Based Molecular Descriptions. *Quant. Struct. Relationships* **21**, 598–604 (2002).
304. Esposito, C., Landrum, G. A., Schneider, N., Stiefl, N. & Riniker, S. GHOST: Adjusting the Decision Threshold to Handle Imbalanced Data in Machine Learning. *J. Chem. Inf. Model.* **61**, 2623–2640 (2021).
305. You, J., Liu, B., Ying, R., Pande, V. & Leskovec, J. Graph convolutional policy network for goal-directed molecular graph generation. in *Advances in Neural Information Processing Systems* (2018). doi:10.48550/arXiv.1806.02473.
306. Jin, W., Barzilay, R. & Jaakkola, T. Multi-Objective Molecule Generation using Interpretable Substructures. in *International Conference on Machine Learning* (2020). doi:10.48550/arXiv.2002.03244.
307. Jasial, S., Hu, Y., Vogt, M. & Bajorath, J. Activity-relevant similarity values for fingerprints and implications for similarity searching. *F1000Research* **5**, 591 (2016).
308. Vass, M. *et al.* Aminergic GPCR-Ligand Interactions: A Chemical and Structural Map of Receptor Mutation Data. *J. Med. Chem.* **62**, 3784–3839 (2019).
309. Khemchandani, Y. *et al.* DeepGraphMolGen, a multi-objective, computational strategy for generating molecules with desirable properties: a graph convolution and reinforcement learning approach. *J. Cheminform.* **12**, 53 (2020).
310. Liu, X., Ye, K., van Vlijmen, H. W. T., IJzerman, A. P. & van Westen, G. J. P. DrugEx v3: scaffold-constrained drug design with graph transformer-based reinforcement learning. *J. Cheminform.* **15**, 24 (2023).
311. Hutchins, D., Schlag, I., Wu, Y., Dyer, E. & Neyshabur, B. Block-Recurrent Transformers. *arXiv* (2022) doi:10.48550/arxiv.2203.07852.
312. Li, Y., Zhang, L. & Liu, Z. Multi-objective de novo drug design with conditional graph generative model. *J. Cheminform.* **10**, 33 (2018).
313. Patronov, A., Margreitter, C., Blaschke, T. & Guo, J. REINVENT 3.0. *GitHub* https://github.com/MolecularAI/Reinvent/tree/reinvent.3.0 (2021).
314. Gentile, F. *et al.* Artificial intelligence–enabled virtual screening of ultra-large chemical libraries with deep docking. *Nat. Protoc.* **17**, 672–697 (2022).

315. Graff, D. E., Shakhnovich, E. I. & Coley, C. W. Accelerating high-throughput virtual screening through molecular pool-based active learning. *Chem. Sci.* **12**, 7866–7881 (2021).
316. Sadybekov, A. A. *et al.* Synthon-based ligand discovery in virtual libraries of over 11 billion compounds. *Nature* **601**, 452–459 (2022).
317. Beroza, P. *et al.* Chemical space docking enables large-scale structure-based virtual screening to discover ROCK1 kinase inhibitors. *Nat. Commun.* **13**, 6447 (2022).