Thesis for The Degree of Licentiate of Engineering

Sequential Decision-Making for Drug Design

Towards closed-loop drug design

Hampus Gummesson Svensson

Department of Computer Science and Engineering
Chalmers University of Technology | University of Gothenburg
Gothenburg, Sweden, 2023

Sequential Decision-Making for Drug Design
Towards closed-loop drug design

Hampus Gummesson Svensson

© Hampus Gummesson Svensson, 2023
except where otherwise stated.
All rights reserved.

ISSN 1652-876X

Department of Computer Science and Engineering
Division of Data Science and AI
Chalmers University of Technology | University of Gothenburg
SE-412 96 Göteborg,
Sweden
Phone: +46(0)31 772 1000

Printed by Chalmers Digitaltryck,
Gothenburg, Sweden 2023.

Till min familj.
To my family.

Sequential Decision-Making for Drug Design
Towards closed-loop drug design

Hampus Gummesson Svensson

Department of Computer Science and Engineering
Chalmers University of Technology | University of Gothenburg

Abstract

Drug design is a process of trial and error to design molecules with a desired response toward a biological target, with the ultimate goal of finding a new medication. It is estimated to be up to 10^{60} molecules that are of potential interest as drugs, making it a difficult problem to find suitable molecules. A crucial part of drug design is to design and determine what molecules should be experimentally tested, to determine their activity toward the biological target. To experimentally test the properties of a molecule, it has to be successfully made, often requiring a sequence of reactions to obtain the desired product. Machine learning can be utilized to predict the outcome of a reaction, helping to find successful reactions, but requires data for the reaction type of interest. This thesis presents a work that combinatorially investigates the use of active learning to acquire training data for reaching a certain level of predictive ability in predicting whether a reaction is successful or not. However, only a limited number of molecules can often be synthesized every time. Therefore, another line of work in this thesis investigates which designed molecules should be experimentally tested, given a budget of experiments, to sequentially acquire new knowledge. This is formulated as a multi-armed bandit problem and we propose an algorithm to solve this problem. To suggest potential drug molecules to choose from, recent advances in machine learning have also enabled the use of generative models to design novel molecules with certain predicted properties. Previous work has formulated this as a reinforcement learning problem with success in designing and optimizing molecules with drug-like properties. This thesis presents a systematic comparison of different reinforcement learning algorithms for string-based generation of drug molecules. This includes a study of different ways of learning from previous and current batches of samples during the iterative generation.

Keywords

Reaction yield prediction, *de novo* drug design, active learning, multi-armed bandits, reinforcement learning

List of Publications

Appended publications

This thesis is based on the following publications:

[**Paper I**] S. Viet Johansson, **H. Gummeson Svensson**, E. Bjerrum, A. Schliep, M. Haghir Chehreghani, C. Tyrchan, O. Engkvist, *Using Active Learning to Develop Machine Learning Models for Reaction Yield Prediction Molecular Informatics 41, 2022, 2200043.* [Contributed in: design of the study, empirical evaluation and analysis, writing the manuscript]

[**Paper II**] **H. Gummesson Svensson**, E. Jannik Bjerrum, C. Tyrchan, O. Engkvist, M. Haghir Chehreghani, *Autonomous Drug Design with Multi-Armed Bandits*
2022 IEEE International Conference on Big Data (Big Data), Osaka, Japan, 2022, pp. 5584-5592. [Contributed in: design of the study, empirical evaluation and analysis, writing the manuscript]

[**Paper III**] **H. Gummesson Svensson**, C. Tyrchan, O. Engkvist, M. Haghir Chehreghani, *Utilizing Reinforcement Learning for Drug Design*
Submitted, under review. [Contributed in: design of the study, empirical evaluation and analysis, writing the manuscript]

Acknowledgment

I would like to express my deepest appreciation to my Ph.D. supervisor, Morteza Haghir Chehreghani, for his invaluable guidance and for constantly challenging me to become a superior researcher. I am also truly grateful to my current and previous industrial advisors Ola Engkvist, Christian Tyrchan, and Esben Jannik Bjerrum for believing in me and sharing their invaluable knowledge. I am thankful to my co-supervisor, Alexander Schliep, and examiner, Graham Kemp, for their important guidance and support.

I am grateful to all of my Ph.D. colleagues at DSAI, for all the laughs and discussions we have had so far and for those that are yet to come, among them Adam, Alexander, Anton, Arman, Christopher, Daniel, David, Deepthi, Emil, Emilio, Fazeleh, Filip, Firooz, Hanna, Hannes, Juan, Lena, Linus, Lovisa, Markus, Mehrdad, Mena, Niklas, Peter, Riccardo, Simon, and Tobias. Many thanks also to all faculty, post-docs, administrators, and everyone else at the division. I am also truly thankful to all of my colleagues at Molecular AI, for creating such a vibrant and friendly workplace, among them Alessandro, Alexey, Annie, Bob, Christos, Emma, Emma, Gökçe, Hannes, Harry, Jiazhen, Jon Paul, Lakshidaa, Lewis, Lili, Marco, Micheal, Mikhail, Pallavi, Peter, Preeti, Rosa, Samuel, Thierry, Thomas, Tomas, Varvara, Vincenzo, and Yasmine.

This work would not have been possible without the support of my family and friends. I am extremely grateful to my best friend and partner, Sophia, for her endless emotional support and for always providing new perspectives. Thanks to my son, Lorentz, for always lighting up the darkest days and for making me understand what is most important in life. I am also truly grateful to my mum, Carola, for her support and for always teaching me new things. I would also like to express my deepest gratitude to my brother, Hannes, for being the best brother one can wish for. Last but not least, I am grateful for the support from all of my friends.

This work was partially supported by the Wallenberg Artificial Intelligence, Autonomous Systems, and Software Program (WASP), funded by the Knut and Alice Wallenberg Foundation, Sweden. Thank you for making this work possible and providing a platform for meeting other ambitious people.

I am truly grateful for having so many smart and helpful people by my side. This work would not have been possible without any of them.

Contents

Part I

Introductory Chapters

Chapter 1

Introduction

Developing a new drug is a complex process that can take up to a decade and cost more than US $1 billion (Paul et al., 2010; Wouters, McKee & Luyten, 2020). Throughout this process, numerous decisions are made, possibly with a large impact on future decisions, requiring informed decisions. A crucial part of this process is to design novel clinical drug candidates with desired molecular properties (Hughes et al., 2011).

Conventional drug design involves human expertise to propose, synthesize and test new molecules. Human experts hold a key position in the decision-making in the design of new drugs, which have so far enabled the finding of thousands of medicinal drugs that both save life and provide better life quality for humans all around the world. It has been estimated that the ensemble of academic, commercial, and propriety chemical databases includes a magnitude of 10^8 existing chemical compounds, while the number of feasible drug-like molecules is theoretically estimated to be between 10^{23} and 10^{60} (Polishchuk, Madzhidov & Varnek, 2013; Reymond & Awale, 2012). Thus, conventional drug design methods seem to concentrate on a relatively small fraction of the chemical space.

Nowadays, machine learning and sophisticated automation of the design of new drugs constitute fundamental strategies to enhance productivity in pharmaceutical research (G. Schneider, 2018; Vamathevan et al., 2019). Significant advances have taken place during the last years in applying machine learning to drug design, in particular with the recent advances of deep learning. One such example is the use of generative models to design molecules that potentially demonstrates a desired set of experimental properties. Instead of a human expert proposing new molecules, these methods leverage deep learning to optimize predicted property values, which represent the set of desired experimental properties. Since their recent introduction, a vast number and variety of generative models have been applied to molecular design (Bilodeau et al., 2022).

Another example of the recent advances in applying machine learning is to assist in the decision of how to synthesize molecules (Coley, Green & Jensen, 2018). For instance, instead of a human expert trying different synthetic routes

for synthesizing a target molecule, machine learning can be used to validate the outcome of a reaction. This requires data for the reaction types of interest, preferably including both unsuccessful and successful outcomes of reactions.

The recent advances in using machine learning can enable a closed-loop drug design platform, where drug molecules are designed in an automated system under human supervision, but no such system has so far been achieved (Bilodeau et al., 2022). For such a system to be achieved, it has to be able to make several decisions on its own, such as, where in the chemical space to focus the search for novel molecules, which molecules to synthesize from the search, and decide how to synthesize the molecules. In fact, it has been argued by experts in the field that a closed-loop platform is necessary for machine learning to make an impact in drug discovery (Saikin et al., 2019).

This thesis studies different sequential decision-making tasks for drug design and is structured in the following way. The first part of the thesis comprises the introductory chapters. Chapter 2 provides an overview of relevant background knowledge to aid in the understanding of the appended papers. This includes an introduction to the phases of drug discovery and design and associated sequential decision-making problems. Followed by a brief introduction to relevant chemoinformatics and *in silico* drug design concepts. It contains a concise introduction to active learning, multi-armed bandit, and reinforcement learning problems. Furthermore, the challenges and consequent research questions considered in this thesis are introduced. Chapter 3 summarizes the problems, methods, results, and contributions of the appended papers. Chapter 4 concludes the main research outcomes of the appended papers and discusses possible future direction. The second part of this thesis comprises the three appended papers.

Chapter 2

Background

This chapter introduces some of the topics and concepts used throughout this thesis.

2.1 Drug Discovery and Design

Figure 2.1: The drug discovery process. UMN, unmet medical needs.

A drug discovery campaign is initiated when a disease with an unmet need for medication has been identified (Hughes et al., 2011). The next step of this campaign is to identify a biological target, e.g., a protein, genes, and RNA, which the drug should interact with and yield a desired response resulting in a therapeutic effect with respect to the identified disease. This refers to the bioactivity (or biological activity), describing the response of a drug on living matter. A molecule with a desired activity is called an active molecule (against the desired biological target). When a target has been identified, it has to be fully validated to gain sufficient confidence in the activity, preferably using multiple validation approaches. After the identification and validation of a biological target, the objective is to screen for molecules that are validated to display the desired response, so-called *hit* molecules. A commonly used screening method is high throughput screening (HTS) which tests a collection of molecules against the validated target, usually conducted in parallel in wells of a microtitre plate by a robotic system (Macarron et al., 2011; Wildey et al., 2017). When several hit molecules have been identified and it has been decided which ones are the most promising, the hits are refined to develop molecules with a larger effect on the biological target, producing so-called lead molecules. When such lead molecules have been developed, they are optimized to further improve their drug-like properties, such as lowering the concentrations needed to obtain a desired response against the biological target. From the optimized

lead molecules, a preclinical candidate along with a backup candidate is usually selected (Vohora & Singh, 2017). This is the end of the drug discovery process (illustrated in Fig. 2.1) and the beginning of the drug development process, with the goal to get the newly discovered drug to the market. The work of this thesis regards drug discovery and in particular drug design.

Drug design has the primary goal of designing a new drug that evokes a desired response, at low concentrations, for a disease and at the same time is free from side effects. Drug design is a vital part of drug discovery, involved in developing a preclinical candidate after the first hit molecules have been identified. The goal is to develop a molecule that is better than the hit molecule. Hence, designing lead molecules and optimized versions of these.

2.1.1 Sequential Decision-Making in Drug Design

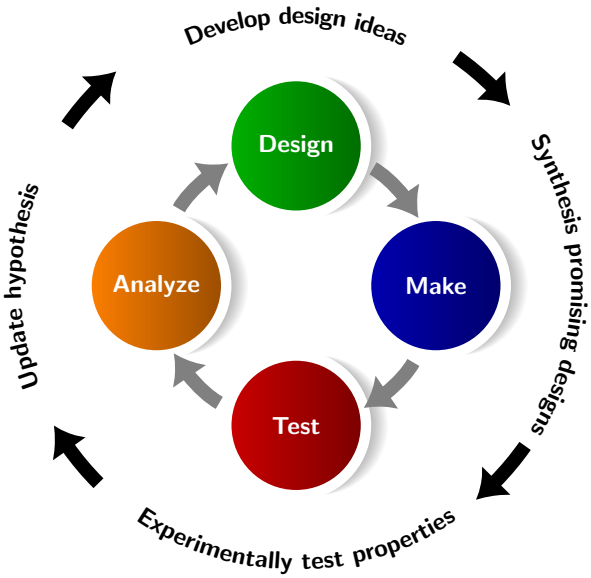


Figure 2.2: The Design-Make-Test-Analyze cycle utilized in drug design.

Drug design is an iterative process involving trial-and-error testing, hence sequential decision-making is a natural part of it. The drug design process is therefore often modeled as the so-called Design-Make-Test-Analyze (DMTA) cycle, illustrated in Fig. 2.2, where one tries to iteratively acquire knowledge to improve the design of novel drugs. In the *Design* step, the goal is to develop molecules that potentially demonstrate the desired experimental properties. Designed molecules are synthesized in the *Make* step and subsequently, if successfully made, experimentally tested to identify their properties. Finally, the acquired knowledge is analyzed and concluded to the next cycle. This is done until a sufficient set of acceptable drug molecules has been designed.

Synthesizing a molecule is not a straightforward task. Synthesis planning is the process by which a chemist or computer decides how to synthesize a molecule. This is usually done by iteratively breaking the desired molecules into intermediates and smaller molecules until reaching an available building block. This is known as retrosynthetic analysis. After identifying possible synthetic routes and building blocks, they have to be verified and further optimized to yield sufficient quality of the desired molecule. High-throughput experimentation (HTE) is a workflow to run multiple reactions in parallel that is nowadays widely used to explore and validate different reaction mechanisms and reaction parameters to obtain an acceptable amount of product (Mennen et al., 2019). This offers a way of trial-and-error testing of reactions to obtain the desired molecule.

2.2 Cheminformatics

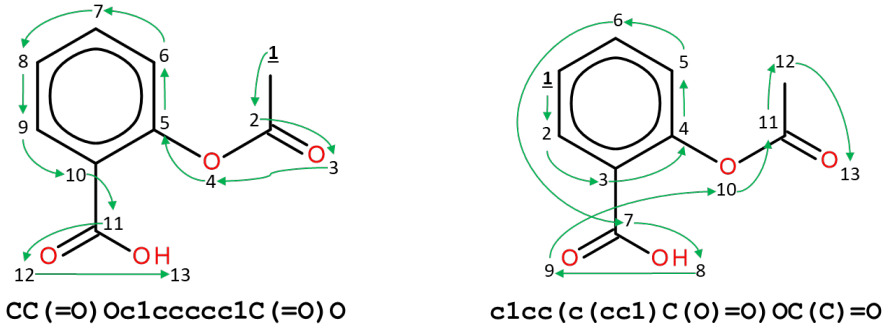
This section introduces relevant computer-readable representations of molecules, and how to analyze and compare molecular structures.

2.2.1 Molecular Representation

Several different molecular representations are used for machine learning in drug design (David et al., 2020). To provide a relevant background for the included papers, we focus on two of them: the simplified molecular-input line-entry system (SMILES) (Weininger, 1988) and molecular fingerprints. SMILES is one among several string-based representations, while molecular fingerprints are vector-based encodings. Many molecular representations (e.g., SMILES strings) are based on the molecular graph representation, where the nodes and vertices in a labeled graph correspond to the atoms and bonds, respectively, of a molecule. The label of each node corresponds to the atom type of the atom represented by that node, while the label of each vertex corresponds to the corresponding bond type. There are several other graph representations but they are not considered in this thesis.

2.2.1.1 Simplified Molecular-Input Line-Entry System (SMILES)

The simplified molecular-input line-entry system (SMILES) is a commonly used line notation system, representing the 2-dimensional molecular graph as a linear string of characters (Weininger, 1988). Firstly, each atom in the molecule is assigned a unique number (hydrogen atoms are normally omitted). Subsequently, the SMILES representation is obtained by traversing the molecular graph in the order given by the unique numbers, appending each traversed atom and non-single bond to the string. The unique number of each atom can be assigned in different ways, leading to different atom orderings in the string representation, as illustrated in Fig. 2.3. For canonical SMILES representation, the ordering is computed to give unique a SMILES representation for the same molecule; while for the randomized SMILES representation, the first atom is randomly assigned and then traverses the graph starting from this atom.



(a) Canonical representation

(b) Randomized representation

Figure 2.3: Canonical and randomized SMILES representation of Aspirin. The canonical representations assign a canonical ordering of the atoms, to provide a unique string representation for each molecule. The randomized representation assigns a random initial atom and then traverses the molecular graph starting at the corresponding node. Figure extracted, with permission, from original work by (Arús-Pous et al., 2019).

2.2.1.2 Molecular fingerprints

Molecular fingerprints are binary or occurrence vectors containing indexed elements encoding the molecular structure (David et al., 2020; Mauri, Consonni & Todeschini, 2017). The molecular structure is explored in all possible substructure patterns by following a pre-defined set of rules, where different types of fingerprints are defined by different sets of rules. A major type of patterns is atom-based patterns. Atom-based patterns exhaustively explore circular patterns around each heavy atom, where the radius of the circular patterns is incremented up to a pre-defined radius. Because of this, fingerprints computed using atom-based patterns are known as circular fingerprints.

A well-known family of circular fingerprints is the extended-connectivity fingerprints (ECFPs), which are obtained by utilizing an algorithm based on the Morgan algorithm (Morgan, 1965; Rogers & Hahn, 2010). The ECFP generation process consists of three sequential stages, as described by Rogers and Hahn (2010): (1) each atom is assigned an integer identifier, e.g., their atomic number, but ignoring hydrogen atoms and bonds. Subsequently, these initial atom identifiers are collected into an initial fingerprint set; (2) each atom identifier is iteratively updated to reflect the identifiers of each atom's neighbors. This is done by each atom collecting its own and immediate neighbors' identifiers into an array and, subsequently, applying a hash function to produce a new integer identifier. Note that the size of the space of identifiers depends on the output size of the hash function. The old atom identifiers are thereafter replaced by the new identifiers, and the new identifiers are added to the fingerprint set. The number of iterations of this procedure is determined by the prespecified radius of this circular fingerprint; (3) duplicate identifiers in the fingerprint set are removed and the final set defines an ECFP fingerprint. Alternatively, the

duplicate identifiers can be kept and hence keep information about multiple occurrences, providing a final fingerprint set that defines an ECFP fingerprint with counts. To be used in practice, the remaining identifiers are usually represented by a vector, e.g., where identifier x implies that bit x is active (1) in the vector, optionally including counts of multiple occurrences.

To evaluate how similar two molecules are, it is common to compute the similarity (or dissimilarity) between their corresponding fingerprint sets. A commonly used similarity metric is the Jaccard (or Tanimoto) coefficient. For two finite sets A and B, e.g., set of bits, the Jaccard coefficient is defined as

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}, \tag{2.1}$$

which gives a value between 0 and 1. As a result, the dissimilarity between two sets, the so-called Jaccard distance, is obtained by $d_J(A, B) = 1 - J(A, B)$. Identical fingerprints do not imply that the corresponding molecules have identical structures since different characteristics in the molecular structure can lead to the same bit being active.

2.2.2 Scaffold Analysis

The scaffold of a molecule is defined as its core structure. This is a common structure characterizing a group of molecules. This provides a basis for a systematic investigation of molecular core structures and building blocks. A popular approach for deriving molecular scaffolds from molecules was formulated by Bemis and Murcko in 1996, therefore known as the Bemis-Murcko scaffold (Bemis & Murcko, 1996). It identifies side chain atoms in the graph representation of a molecule and removes these from the graph, as illustrated in Fig. 2.4. It is also common to derive a more generic scaffold to analyze the topological relationships between molecules. A topological scaffold can be derived by the Bemis-Murcko scaffold, e.g., by converting all atom types into carbon atoms and all bonds into single bonds.

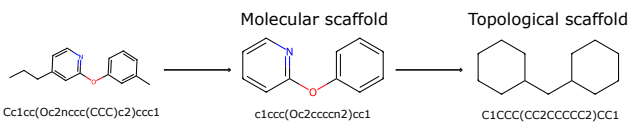


Figure 2.4: The structural formula and SMILES strings for an arbitrary molecule, and its molecular scaffold and a corresponding topological scaffold based on the Bemis-Murcko algorithm.

When the core structure of a molecule has been derived, it can be used to find structurally distinct molecules having similar activity, known as *scaffold hopping* (Hu, Stumpfe & Bajorath, 2016). This can aid in providing several structural alternatives when designing drug molecules, e.g., utilizing computational (virtual) screening approaches.

2.3 *in silico* Drug Design

Drug design relies more frequently on computational methods, also known as *in silico* methods, to generalize knowledge to unseen molecules and aid the human expert in making informed decisions (P. Schneider et al., 2020). This section presents some of the key computational concepts and methods in drug design.

2.3.1 Quantitative Structure-Activity Relationship (QSAR)

Quantitative structure-activity relationship (QSAR) methods aim to predict a molecule's chemical bioactivity and physical properties, given its structure (Tyrchan et al., 2022). In fact, they are regression or classification models that seek to learn a relationship between a complex property and observable, so-called descriptors. These models are based on the principle that similar molecules tend to have similar properties (Bender & Glen, 2004). The descriptors encode structurally derived properties from both the 2D and 3D structure of a molecule, such as topological, geometrical, or electronic features. For instance, molecular fingerprints can be used to encode relevant structural properties. These models are often used for the computational (virtual) screening of millions of compounds to reduce the number of candidates to be synthesized and tested experimentally, ultimately speeding up the identification of possible drug candidates (Neves et al., 2018). Random forest models are still considered the standard for QSAR methods, but gradient boosting and deep learning methods are nowadays popular alternatives (Muratov et al., 2020).

Logically, the descriptors should be selected to represent the molecular features relevant to the properties of interest (Danishuddin & Khan, 2016). Hence, setting up a QSAR model requires careful consideration of both experimental errors in the data and generalization errors of the model. Experimental errors can be caused by errors in the chemical structures in the data, while generalization error is possibly caused by an insufficient relationship between the descriptor of the considered molecular properties and the response variables (Tropsha, 2010). QSAR models utilize descriptors of the chemical structure and small errors in the chemical structure can lead to a significant reduction of predictive ability (Young et al., 2008).

2.3.1.1 Inverse QSAR

In the inverse QSAR problem, the aim is to identify a molecular structure fulfilling desired properties (Tyrchan et al., 2022). This makes it a suitable problem for molecular *de novo* design, described in Section 2.3.2, which recently has gained a lot of attention. A fundamental problem for inverse QSAR is that molecular descriptors are not continuous and unique, while the same holds for the property space. Hence, molecules with the same descriptor can display different properties, while molecules with different descriptors can display similar properties.

2.3.2 *de novo* Drug Design

The fundamental goal of molecular *de novo* design is to identify novel chemical structures that satisfy a set of predefined criteria (Tyrchan et al., 2022). This can be formulated as an optimization problem where the objective is to find a molecular structure that optimizes the ground truth property values, which are represented by the pre-defined criteria. *de novo* drug design refers to this problem for drug design. As a result of the recent progress in machine learning, especially in deep learning, generative models are now widely used in *de novo* drug design to traverse the chemical space (Meyers, Fabian & Brown, 2021). The goal is that these models should learn to effectively identify chemical structures in the chemical space that fulfills the predefined criteria. A vast number and variety of different machine learning techniques have been used for this, including techniques such as genetic algorithms, monte-carlo tree search, variational autoencoders, Bayesian optimization, and reinforcement learning (Gao et al., 2022; Sanchez-Lengeling & Aspuru-Guzik, 2018; Thomas, O'Boyle et al., 2022). Another aspect to consider is the use of a suitable molecular representation, e.g., string-based representations and molecular graphs. The work of this thesis focus on SMILES-based *de novo* drug design utilizing reinforcement learning to optimize the predefined criteria.

2.3.3 Computer-Aided Synthesis Planning

There are two main problems in computer-aided synthesis planning: synthetic route prediction (retrosynthesis) and forward prediction (Johansson et al., 2019). The former problem aims to predict synthesis routes and building blocks necessary to synthesize a specific molecule. The latter problem tries to predict the reaction outcomes, given building blocks (reactants) and reaction conditions (e.g., temperature, solvent, and catalyst). Hence, such a predictive model can be used to validate that the desired product is produced by the proposed reaction and suggest suitable reaction conditions (Schwaller et al., 2019). Synthetic route prediction algorithms are usually either template-based or template-free (Segler & Waller, 2017; Zheng et al., 2019). Template-based algorithms follow manually encoded chemical transformation rules of known reactions, while template-free algorithms are not constrained to follow such transformation rules. A problem related to forward prediction is reaction yield prediction where the objective is to predict the yield of a reaction (Schwaller et al., 2021). The reaction yield describes the quantity (usually in percentage) of the building blocks that are converted to the desired product(s) in the reaction. This is normally done by either explicitly predicting the reaction yield or predicting if the yield will reach a desired quantity. The latter is of more interest in drug design where the objective is to find a successful synthetic route to experimentally test properties; while the former is usually of more interest when preparing for drug development since the desired product then needs to be manufactured in a sufficient quantity. The work of this thesis considers reaction yield prediction for predicting if a reaction will successfully provide a desired minimum reaction yield.

2.3.4 Data Curation

Modern methods often require a vast amount of data, especially with the recent rise of deep learning. Large datasets utilized in drug design originate from various sources, such as ChEMBL consisting of extracted and manually curated structure-activity relationship (SAR) data from the primary medicinal chemistry and pharmacology literature (Gaulton et al., 2012). Hence, data curation is an important aspect of machine learning in drug design. Data curation includes several steps of cleaning and standardization of the chemical data, such as the removal of mixtures, inorganics and salts, and standardization of chemical structure and bioactivity data (Tropsha, 2010). It also includes the removal of duplicates and treatment of tautomeric forms. Tautomers of a molecule only differ by an intramolecular movement of a hydrogen atom from one atom to another. Tautomers usually have different molecular fingerprints and other properties, such that similar molecules encoded as different tautomers are unintentionally considered, which can influence the predictive ability (Martin, 2009; Masand et al., 2014). Removal of duplicates can be accomplished by standardizing the representation of the chemical structure, e.g., canonicalization of SMILES string, and removing all chemical structures with the same representation. In addition, descriptors calculated from a 2D representation will usually recognize molecules with minor differences in 3D structure (e.g., molecules that are mirror images of each other) as duplicates (Tropsha, 2010).

2.4 Active Learning Problems

In supervised learning, a learner chooses a mapping between data instances $\mathcal{X}$ and labels $\mathcal{Y}$, with the objective of outputting a desired label $y \in \mathcal{Y}$ given a data instance $x \in \mathcal{X}$. A suitable mapping is usually decided by using a training set $\mathcal{L} = \{(x, y)^{(l)}\}_{l=1}^{L} \subset \mathcal{X} \times \mathcal{Y}$ of tuples of a data instance and a desired label to output. To construct such a training set, for each data instance of interest, the desired label needs to be acquired. In some cases, these labels can be easily obtained or are already available; while in other cases these labels can be more difficult to obtain. *Active learning* concerns how to improve the generalization of the learning by utilizing a carefully chosen training set. The learner then sequentially decides on what label(s) to query from an oracle, e.g., a human annotator, by utilizing the information acquired up to this point. On the other hand, in *passive learning* the learner has no control over the training set, e.g., what data instance to query is chosen randomly.

Two common active learning scenarios are stream-based and pool-based active learning (Settles, 2012). In stream-based active learning, the learner gets prompted with one unlabelled data instance at a time and has to immediately choose between two options: keep the data instance and query its label, or discard the data instance. This is in contrast to pool-based active learning, where the learner has access to a pool of unlabelled data instances. At each iteration, the learner needs to decide on what data instance(s) to add to the training set from the pool, and consequently query the label from the oracle.

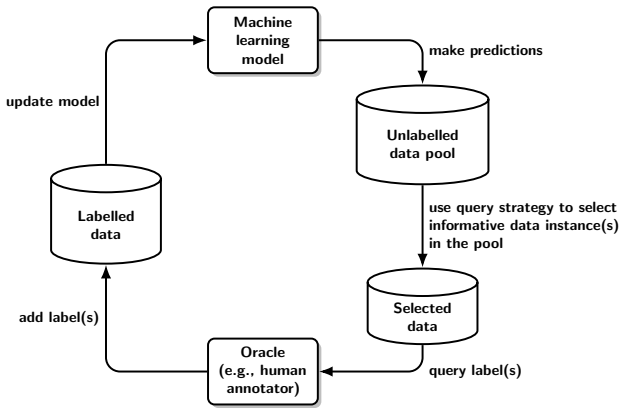


Figure 2.5: Pool-based active learning.

The strategy used to decide on what label(s) to query is known as the *query strategy*. A well-known and established type of query strategy is uncertainty sampling (Schein & Ungar, 2007; Yang et al., 2015). The philosophy of this type of strategy is that if the learner is uncertain about the label of an instance, then the label of this instance is more informative for the learner to know, compared to knowing the labels of an instance that the learner is certain about. The work of this thesis focuses on uncertainty sampling in the pool-based active learning setting.

2.5 Multi-Armed Bandit Problems

Imagine going to a casino with M slot machines, also known as one-armed bandits. In each round, you can choose a slot machine to play by pulling its "lever". For each machine, there is a chance of winning a certain payout. Over a total of T rounds, you want to maximize the sum of payouts by identifying the slot machine with the highest average payout. Hence, in each round, you need to decide whether to try a new machine, play a machine that you have only tried a few times or play the machine that has given the highest average payout so far. This is known as the multi-armed bandit (MAB) problem, first discussed by Robbins (1952) and later formalized by Lai, Robbins et al. (1985). It models the exploitation-exploration dilemma, described above, where in each round one has to determine whether to keep exploring new alternatives or be satisfied with the current best alternative. A naïve strategy to tackle this dilemma is to choose the action with the largest empirical expected outcome with a probability ϵ and otherwise choose a random action. This is known as the epsilon-greedy strategy (ϵ-greedy), where the greedy strategy (i.e., $\epsilon = 1$) always chooses the best action in hindsight. When choosing the best action in hindsight we are exploiting our current knowledge, while when choosing a random action we are exploring to learn more about the general outcomes of actions.

Algorithm 1 The multi-armed bandit problem

Input: time horizon T, policy π
Initialization: history $H_0 = \emptyset$
for $t = 1, \dots, T$ **do**
  Choose and perform action $a_t \sim \pi(\cdot|H_{t-1})$
  Observe reward r_t
  Update history $H_t \leftarrow (H_{t-1}, (a_t, r_t))$
end for

The multi-armed bandit problem is a sequential game between a learner and an environment where the learner tries to learn probable outcomes of the environment for different actions. In general, it is possible to model numerous sequential decision-making problems as a MAB problem, extending the original problem, e.g., the design of clinical trials, news recommendation, finance, navigation, and bottleneck identification (Åkerblom, Chen & Chehreghani, 2020; Åkerblom, Hoseini & Chehreghani, 2022; Li et al., 2010a; Press, 2009; Shen et al., 2015; Villar, Bowden & Wason, 2015).

The problem is formally defined as follows. In each round $t \in [T]$, a learner chooses an action a_t from a set $\mathcal{M}$ of M possible actions, also known as *arms*. Subsequently, the learner observes a reward $r_t \in \mathbb{R}$ from the environment. The learner decides on what action a_t to choose, based on the history $H_{t-1} = ((a_1, r_1), \dots, (a_{t-1}, r_{t-1})) \in (\mathcal{M} \times \mathbb{R})^{t-1}$ of previous actions and rewards, using a mapping from histories to actions — a *policy*. The most common objective for the learner is to learn a policy to optimize the total cumulative reward of the learner $\sum_{t=1}^{T} r_t$. This is done for an unknown environment where the learner only knows that the environment is part of environments class $\mathcal{E}$, i.e., a set of possible environments.

One type of environment class is the stochastic MAB problem, so-called stochastic bandits, where the reward of each action is drawn independently from a fixed probability distribution with prior unknown parameters, e.g., a Bernoulli distribution with unknown parameters. For each possible action $a \in \mathcal{M}$ there exists an unknown expected value μ_a that (partially) determines the reward distribution for that action. To optimize the cumulative reward of the learner, the objective is to identify the best action $a^* = \text{argmax}_{a \in \mathcal{M}} \mu_a$, i.e., the action with the highest expected value. We wish to do this in the least number of rounds, usually with the aim to minimize the regret over all each round

$$R(T) = \mu^* - \mathbb{E}\left[\sum_{t=1}^{T} r_t\right] = \mu^* - \sum_{t=1}^{T} \mu_{a_t}, \tag{2.2}$$

where μ^* is the true expected reward of the optimal arm. Hereafter, for the relevance of this thesis, we focus on stochastic bandits and refer to the work by Slivkins (2022) and Lattimore and Szepesvári (2020) for a comprehensive overview of different extensions of the original MAB problem. Below follows a brief introduction to some more advanced types of stochastic MAB problems, relevant to this thesis, based on the introduction in appended Paper II.

2.5.1 Contextual Bandits

In the contextual MAB problem, before choosing which arm to play in the current round, the learner observes a feature vector, known as the *context*. The reward of each round is assumed to depend on both the observed context and the chosen action. The contextual MAB problem has been broadly studied under the linear realizability assumption, introduced by Abe, Biermann and Long (2003), where the expected reward is assumed to be linear with respect to the context vector of each arm (S. Agrawal & Goyal, 2013; Auer, 2002; Chu et al., 2011; Li et al., 2010b). There have been several successes in using the contextual MAB problem to model real-life applications, such as recommender systems, health applications, and information retrieval (Bouneffouf, Rish & Aggarwal, 2020).

2.5.2 Multiple-Play Bandits

Up to this point, we have assumed that the learner only chooses one arm ($K = 1$) in each round. Allowing the learner to choose more than one arm in each round ($K > 1$) is known as the multiple-play MAB problem (R. Agrawal, Hegde, Teneketzis et al., 1990; Komiyama, Honda & Nakagawa, 2015), first introduced by (Anantharam, Varaiya & Walrand, 1987). In this problem, a super arm consisting of a combination of $K \leq M$ base arms $A \subseteq \mathcal{M}$ is played in each round. In the multiple-play problem, all combinations of (unique) K base arms are usually allowed and a reward is observed for each individual base arm. However, there are other similar problems (e.g., the combinatorial MAB problem) where not all combinations of base arms are allowed and the reward of each individual arm is not necessarily observed (instead the sum of rewards of all chosen base arms is possibly observed).

2.5.3 Sleeping and Volatile Bandits

The standard MAB problem assumes that there is a fixed set $\mathcal{M}$ of M available arms in each round. However, in real-life applications, it is possible that the set of available arms differs between rounds, e.g., a slot machine is occupied by another player for some rounds. Hence, in each round t there is a set $\mathcal{M}_t \subseteq \mathcal{M}$ of available arms in this round. This setting is studied by Kleinberg, Niculescu-Mizil and Sharma (2010) by introducing *sleeping bandits*. In each round, the set of available arms in each round is chosen from a fixed and finite pool of actions by an adversary. They propose an algorithm that prioritizes playing an arm that has become available for the first time. Otherwise, it plays the arm with the largest upper confidence bound, inspired by the UCB1 algorithm (Auer, Cesa-Bianchi & Fischer, 2002). The volatile MAB problem is a similar problem but does not necessarily restrict the problem to a finite pool of arms (Bnaya et al., 2013). Both variants consider *volatile* arms that can “appear” and “disappear” in each round.

2.5.4 Bandits With Similarity Information

Although an extensive collection of MAB algorithms for problems with a fixed small number of arms have been proposed in the literature, MAB problems with infinite or exponentially large arm sets are relatively little studied. For such a problem, one common approach is to use similarity information (or a metric) between contexts and/or arms, by assuming that similar actions yield similar rewards.

For instance, Kleinberg, Slivkins and Upfal (2008) introduce the *Zooming algorithm*, where the similarity information is given as a metric space of arms Kleinberg, Slivkins and Upfal, 2019. Their algorithm tries to approximately learn the expected rewards over the metric space by probing different "regions" of the space, which leads to an adaptive partitioning of the metric space (Slivkins, 2022). At each round t, there is a set of active arms, determined by an activation rule. Each active arm x covers a region of the metric space. This region is given by the confidence ball of the arm $B(x, r_t(x))$, which is a ball with the arm at its center. The radius of the ball is the confidence radius $r_t(x)$ of the empirical average reward (of the active arm) at round t. The confidence radius is related to the size of the one-sided confidence interval of the empirical average reward and guarantees, with high probability, that the difference between the true expected reward and empirical average reward is not larger than the confidence radius. To determine what active arm to play, it chooses an arm with the largest upper confidence bound, similar to the arm selection of UCB1 algorithm (Auer, Cesa-Bianchi & Fischer, 2002).

Slivkins (2011) extends the Zooming algorithm to the contextual setting, where the similarity information is provided by a metric space of context-arm pairs. The work of this thesis extends the techniques developed in this work to allow volatile arms and multiple-play. We relax the contexts and define the space of arms by their corresponding feature vectors.

2.6 Reinforcement Learning Problems

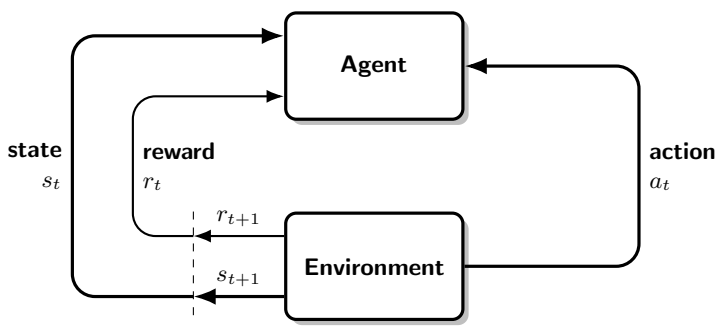


Figure 2.6: Interaction between the agent (learner) and environment in reinforcement learning problems.

A more general type of sequential decision-making problem is the reinforcement learning problem, which is extensively described and overviewed by the work of Sutton and Barto (2018) and Szepesvári (2010). In each round t, a learner observes a state $s_t \in \mathcal{S}$ of the environment and given this state chooses an action $a_t \in \mathcal{M}$ using a policy $\pi(\cdot|s_t) : \mathcal{S} \mapsto \mathcal{M}$. The policy is a learnable mapping from states to a possible action, often providing probabilities of each possible action. Subsequently, a reward $r_{t+1} \in \mathbb{R}$ and new state $s_{t+1} \in \mathcal{S}$ is observed by the learner. The objective of the learner is to learn an optimal policy. The optimality is usually measured in terms of maximization of the discounted future reward, defining the (infinite) return G_t following round t

$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}, \tag{2.3}$$

where γ is the discount factor, $0 \leq \gamma \leq 1$, which is used to penalize the uncertainty of future rewards. For an infinite number of rounds, the return could itself be infinite, which is not desired since we want to maximize it. This can be handled using the discounted cumulative reward where $\gamma < 1$ holds. Multi-armed bandits problems, as described above, can be seen as reinforcement learning problems with only one state, but the history $H_{t-1} = ((a_1, r_1), \ldots, (a_{t-1}, r_{t-1}))$ of previous actions and rewards can also be considered as the *information state* of the problem.

Reinforcement learning problems are usually assumed to be described by a Markov decision process (MDP). A Markov decision process is described by a tuple $(\mathcal{S}, \mathcal{M}, P_a, \gamma)$. As introduced above, $\mathcal{S}$ is the set of states, $\mathcal{M}$ is the set of possible actions and γ is the discount factor. Furthermore, $P_a(r, s, s') = \Pr(s_{t+1} = s', r_{t+1} = r | s_t = s, a_t = a)$ is the probability of observing s' and r as the next state and reward, respectively, when at state s and performing action a. The state transitions of the MDP satisfy the Markov property since, given the current state s and action a, the probability of the next state s' and reward r is independent of all previous states and actions.

The expected return following a policy π starting at state s is known as the *value* $v_\pi(s)$ and is defined as follows

$$v_\pi(s) = \mathbb{E}_\pi \left[G_t | s_t = s \right] = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s \right]. \tag{2.4}$$

In the same way, the expected return of taking action a at state s and subsequently following a policy π is known as the *action-value*, and is defined as follows

$$q_\pi(s, a) = \mathbb{E}_\pi \left[G_t | s_t = s, a_t = a \right] = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a \right]. \tag{2.5}$$

$v_\pi(s)$ and $q_\pi(s, a)$ are called the state-value function and state-action function (for policy π), respectively.

There are two main classes of algorithms to solve reinforcement learning problems: model-free and model-based algorithms. Model-based algorithms

either learn or have access to a model of the environment (Anthony, Tian & Barber, 2017; Chua et al., 2018; Kaiser et al., 2020; Silver et al., 2017). By utilizing a model of the environment that infers the state transitions and rewards, it is possible to learn without interacting with the environment, reducing the sample complexity. Model-free algorithms do not utilize a model of the environment, instead, they only learn using experiences by interacting with the environment. There are two main types of model-free algorithms: policy-based and value-based algorithms. The objective of value-based algorithms is to learn the state-action function of the optimal policy and use the learned state-action function to interact with the environment. If the state-action function of the optimal policy is known, it is possible to determine the optimal action at each state, learning a deterministic policy. Value-based algorithms usually use a more explorative policy to interact with the environment, known as the behavior policy, compared to the policy we want to learn, the so-called target policy. For policy-based algorithms, the policy is directly learned by parameterizing the policy. This is also known as *policy optimization* since the objective is to optimize the policy to maximize the return. Learning the policy directly enables the agent to directly build a stochastic policy and is especially more efficient in continuous action and/or state space, where a continuous policy can be directly learned. To (explicitly or inexplicitly) learn a policy from experience gathered by another policy is known as off-policy learning. On the contrary, in on-policy learning, the same policy is used to gather experiences from the environment and, subsequently, is updated using these experiences. Off-policy learning uses different target and behavior policies, while the same policy is used for both in on-policy learning.

To scale to problems with large action and/or state spaces, it is common to use neural networks as function approximators. Neural networks can approximate a wide range of functions (Hornik, Stinchcombe & White, 1989). Most modern reinforcement learning algorithms use deep neural networks to learn policies, value functions, and models that generalize to unobserved or rarely observed state-action pairs (Arulkumaran et al., 2017; Lillicrap et al., 2015; Silver et al., 2017). This gives rise to the term deep reinforcement learning consisting of reinforcement learning algorithms using deep learning. Several recent deep reinforcement learning algorithms utilize a technique called experience replay, where past experiences are stored in a replay buffer and are replayed during the learning phase (Silver et al., 2017; Wang et al., 2016). Hence, the learning is averaged over its previous states and actions. This provides a way to remove correlations of trajectories used for update and not forget possibly rare experiences, ultimately stabilizing the learning (Schaul et al., 2015).

The focus of the work of this thesis lies in policy-based algorithms utilizing neural networks. In particular, this thesis considers policy optimization algorithms based on Proximal Policy Optimization (PPO), Advantage Actor-Critic (A2C), Soft-Actor Critic (SAC), Actor-Critic with Experience Replay (ACER), and REINFORCE (Haarnoja et al., 2018; Mnih et al., 2016; Schulman et al., 2017; Wang et al., 2016; Williams, 1992).

2.7 Research Challenges and Questions

In this section, we describe three challenges that motivate our research based on the gaps in the literature. Furthermore, we formulate six research questions based on these challenges.

2.7.1 Reseach Challenges

This thesis seeks to address the following research challenges (Qs), which are motivated by gaps in the literature:

C1 Unclear performance gain in using active learning for reaction yield prediction.

For reaction yield prediction, active learning still struggles to show a significant performance gain compared to randomly selecting data points to query, so-called random sampling, when only a few data points have been labeled (Eyke, Green & Jensen, 2020). Existing work on active learning investigates how the performance of active learning depends on conditions such as the initial size of labeled data and capacity of neural network (Bossér, Sörstadius & Chehreghani, 2021). However, this can be task-dependent and no existing work studies the effect of different conditions for active learning on the reaction yield prediction task.

C2 No existing approach considers what to make next based on suggestions developed by *de novo* drug design.

There has been a large focus on *de novo* drug design to optimize a fixed objective function in the Design step of the DMTA cycle (Meyers, Fabian & Brown, 2021). However, to our knowledge, no previous work has considered the sequential decision of which molecules to make next in the DMTA cycle when the design objective is iteratively updated. This a problem that should be addressed to enable closed-loop drug design.

C3 There is no systematic comparison of how the sample efficiency of reinforcement learning-based *de novo* drug design is affected by iteratively learning from a subset of current and/or previous samples.

For *de novo* drug design utilizing reinforcement learning, several existing work studies approaches using a Hill-Climb algorithm to learn from a subset of previous samples (Brown et al., 2019; Neil et al., 2018; Thomas, O'Boyle et al., 2022). However, to our knowledge, no existing work in this domain systematically studies approaches to utilize current and previous samples.

2.7.2 Research Questions

Motivate by the above challenges, we formulate the following research questions (RQs) which are considered in this thesis:

RQ1 *How is the performance of active learning for reaction yield prediction affected by different conditions?*

To tackle challenge (C1), we propose to study how active learning performs under different conditions, compared to passive learning (e.g., random sampling), such as initial sizes of labeled data, machine learning algorithms, and reaction datasets.

RQ2 *How many data instances are needed to be queried by active learning, compared to passive learning, when the objective is to reach a certain level of predictive ability on a reaction yield prediction task?*

In real-life applications, such as reaction yield prediction, a predictive model should have a sufficiently "good" predictive ability on a validation/test set to be usable. Hence, to tackle challenge (C1), we also propose to investigate how much training data is needed when using active learning to reach a certain level of predictive ability, compared to passive learning (e.g., random sampling).

RQ3 *Can the problem of what to make next in the DMTA cycle be formulated as a multi-armed bandit problem?*

To tackle challenge (C2), we propose to find a solution by formulating the problem as a multi-armed bandit problem, which is introduced in Section 2.5.

RQ4 *Can the contextual Zooming algorithms be extended to provide a solution to the problem of what to make next in the DMTA cycle? How should it select arms covered by the same ball?*

To tackle challenge (C2), we propose to extend the contextual Zooming algorithm (Slivkins, 2011) to the formulated multi-armed bandit problem of what to make next in the DMTA cycle. Also, we consider how the extended version should distinguish between arms covered by the same balls to improve the sample efficiency and novelty.

RQ5 *How can string-based de novo drug design utilizing reinforcement learning improve its sample efficiency by learning from a subset(s) of previous and current samples? How does it compare with using all samples in the current round for learning?*

To tackle challenge (C3), we propose to systematically investigate different approaches for learning from a subset(s) of previous and current samples. This should obviously be compared with learning from the full set of samples in the current round.

RQ6 *For string-based de novo drug design, how does the reinforcement learning algorithm affect the sample efficiency when learning from a subset(s) of previous and current samples?*

To tackle challenge (C3), we propose to systematically study different reinforcement learning algorithms when learning from a subset(s) of previous and current samples.

Chapter 3

Summary of Included Papers

In this chapter, the three papers included in this thesis are summarized, including the research contributions. All papers concern sequential decision-making for drug design. Paper I investigates the use of active learning to improve reaction yield prediction. Paper II formulates what to make next in the DMTA cycle as a multi-armed bandit problem and suggests an algorithm for solving this problem. Paper III systematically investigates different deep reinforcement learning algorithms and replay buffers for SMILES-based *de novo* drug design.

3.1 Paper I

In Paper I, we investigate the use of active learning to iteratively improve machine learning models for predicting a reaction to be either successful or unsuccessful, i.e., binary reaction yield prediction of if a reaction will obtain a sufficiently high percentage yield. Given an initial set of labeled reaction data, we iteratively query the label of an unlabelled data point and subsequently retrain the model utilizing the newly acquired label. The objective is to, under different conditions, study how the predictive performance is affected by using active learning for querying labels, and the relative change in the amount of training data needed to achieve different levels of predictive ability.

For the task of predicting whether a reaction will be successful or not, we compare a neural network with a single hidden layer, a neural network with three hidden layers, a Bayesian matrix factorization model, and a random forest model. For these models, we evaluate active learning by utilizing a well-known uncertainty sampling approach based on the output margin. We compare this strategy with random sampling, i.e., passive learning where which label to query is chosen randomly. Also, we investigate how the size of the initial pool of labeled data affects the predictive ability. We evaluate the use of active learning on two fully combinatorial data sets of two different reaction types,

with different reaction variables varied. Hence, this is a retrospective study where all true labels (a successful reaction or not) are known a priori but in our setting we assume that they are unknown until they have been queried.

One-hot encodings are utilized to study how the predictive ability is affected by active learning when only trying to learn the combinatorial patterns in the data. Our findings suggest a relationship between how well the machine learning models have learned the observed reaction data and how large the positive impact of active learning is on predictive performance. We also conduct a feature importance analysis that provides further indications of this. Furthermore, the better predictive ability we require the models to have, the larger gain in using active learning is observed.

Contributions Hampus Gummesson Svensson and Simon Viet Johansson equally performed the main work, and the work was jointly supervised by Morteza Haghir Chehreghani, Ola Engkvist, Esben Jannik Bjerrum, Alexander Schliep and Christian Tyrchan.

3.2 Paper II

Paper I seeks to benefit the problem of how to synthesize a molecule to be able to experimentally test its properties. Prior to this problem, it has to be decided which molecules to make next. In paper II, we formulate the problem of which molecules to make next, a decision made between the Design and Make step of the DMTA cycle, as a stochastic multi-armed bandit problem. In a potential closed-loop drug design platform, *de novo* drug design can be utilized in the Design step to generate a large set of molecules that optimizes a function that scores each generated molecule, the so-called scoring function. The goal is then to update the parameters of the scoring function, to provide a more precise molecular generation, by using experimental data. However, experimental data is both costly and time-consuming to acquire, and therefore it is not possible to make, test and analyze all of the generated molecules. Naïvely one could acquire experimental data for the top-scoring molecules, but this is not necessarily the best approach.

To formulate this as a stochastic multi-armed bandit problem we consider a setting with multiple-plays and volatile arms. The multiple-play setting is appropriate since it should be possible to select several molecules to make, test and analyze in parallel before designing new molecules. The volatile arms setting is considered because a completely new set of molecules can be generated in every design step, due to the randomness in the generation and the iterative update of the scoring function.

To solve this bandit problem, we propose a Zooming algorithm with multiple plays and volatile arms, which extends the contextual Zooming algorithm by (Slivkins, 2011) to our problem. The algorithm partitions the dissimilarity space of feature vectors of each base arm into balls with different radii, where the initial partition consists of a ball covering the entire dissimilarity space.

Given observed rewards for base arms covered by a ball, the empirical mean reward and corresponding confidence radius are computed. If the confidence radius of the empirical mean reward is less than or equal to the radius of the ball, the partition is refined by creating a new ball with half the radius. The radius of the ball is fixed, while the confidence radius is updated when rewards for base arms covered by the ball are observed. A set of available base arms and corresponding feature vectors are observed at the beginning of each round, and subsequently, a super arm of multiple available base arms is chosen. Each base arm is chosen based on its index, which is computed from the mean empirical mean reward, radius, and confidence radius of the ball covering the accompanied feature vector.

For a fixed budget of molecules to be selected, we evaluate the proposed algorithm by comparing it with random selection, selecting the top-scoring molecules with respect to the current scoring function (greedy selection), and a combination of these two (ϵ-greedy selection). We use a dissimilarity space consisting of Morgan fingerprints where the dissimilarity is measured using the Jaccard distance. For the proposed bandit algorithm, to investigate the effect of distinguishing arms covered by the same ball, we study the usage of a weighted index that takes into account the current score of the corresponding molecule. We find that the unweighted variant of our proposed algorithm performs among the best in the early cycles, while the weighted variant performs better in the later cycles. This suggests that utilizing the benefits of both the unweighted and weighted variants can provide the overall best performance.

Contributions Hampus Gummesson Svensson performed the main work, and Morteza Haghir Chehreghani, Ola Engkvist, Esben Jannik Bjerrum, and Christian Tyrchan jointly supervised the work.

3.3 Paper III

The selection of what to make next in Paper II depends on the molecular *de novo* design in the Design step of the DMTA cycle. Ideally, a structurally diverse set of molecules should be generated, to allow for sufficient exploration and exploitation in the selection of what to make next. Previous work has shown promising results using reinforcement learning for molecular *de novo* design, compared to other approaches such as variational autoencoders (Gao et al., 2022; Thomas, O'Boyle et al., 2022). Moreover, several works have proposed to combine reinforcement learning with a Hill-climb algorithm, which learns from the k top-scoring sequences (Brown et al., 2019; Neil et al., 2018; Thomas, O'Boyle et al., 2022).

In Paper III, we investigate various reinforcement learning algorithms for iteratively performing SMILES-based generation of batches of molecules. The policy optimization reinforcement learning algorithms that we investigate in the paper are Proximal Policy Optimization (PPO), Advantage Actor-Critic (A2C), Soft Actor-Critic (SAC), Actor-Critic with Experience Replay (ACER),

and Regularized Maximum Likelihood Estimation (MLE). SAC and ACER are off-policy algorithms developed for off-policy learning, while the others are developed for on-policy learning. All algorithms iteratively update a policy, pre-trained on the ChEMBLE dataset, that provides probabilities over the next character to append in a SMILES string. Multimodal sampling is used to choose the next character given the current policy and the SMILES string is finalized when the stop token is chosen as the next character.

For a pre-defined budget of generated molecules, we compare the number of active molecules and the corresponding number of scaffolds when restricting to seven different ways to learn from sets of molecules generated in the current iteration and previous iteration: (1) learn from the batch of molecules generated in the current iteration; (2) learn from the batch of current molecules and previously generated molecules with diverse rewards; (3) learn from a subset of the current batch with diverse rewards; (4) learn from the current batch and high- and low-rewarding molecules from previous batches; (5) learn from a subset of the current batch that only includes high- and low-rewarding molecules; (6) learn from the current batch and high-rewarding molecules from previous batches; (7) learn from a subset of the current batch that only includes high-rewarding molecules. We collect all these approaches under the term replay buffers, due to their nature to store and provide both current and previously generated molecules. These approaches are inspired by the proposals to combine reinforcement learning with a Hill-climb algorithm. Since the off-policy algorithms SAC and ACER already include an on-policy update step, where the full current batch was utilized, they were compared using only the second, fourth, or sixth replay buffer.

We evaluate the different combinations of policy optimization algorithms and replay buffers for the generation of molecules predicted to be active against the dopamine receptor DRD2. This is evaluated both with and without a diversity filter which penalizes the generation of molecules with similar scaffolds between different iterations. Generally, we find that using at least both high- and low-rewarding molecules is advantageous for generating a large number of active compounds with diverse scaffolds. Using off-policy algorithms with several off-policy updates does not necessarily gain a larger number of active molecules and scaffolds, but displays the potential to display a larger number than the on-policy algorithms.

Contributions Hampus Gummesson Svensson performed the main work, and the work was jointly supervised by Morteza Haghir Chehreghani, Ola Engkvist, and Christian Tyrchan.

Chapter 4

Concluding Remarks and Future Directions

In this thesis, we investigate three sequential decision-making problems for drug design using machine learning. Active learning, multi-armed bandit, and reinforcement learning problems are concerned. The main research result of Paper I was that increasing the training data by utilizing active learning can enhance the predictive ability of reaction yield prediction, compared to training on the same amount of random data. In particular, this was done in the fundamental setting using one-hot encoding and increasing the training data by one point at a time. There is a potential larger gain of using active learning in settings with more elaborate feature vectors for each reaction and when a combination of points are simultaneously added to the training data. However, such a problem is more complex and most likely requires more elaborate active learning methods, e.g., considering the diversity of queried set and uncertainty in feature vectors. In Paper II, the main research outcomes include the formulation of which molecules to make and test next as a multi-armed bandit problem. In addition, an algorithm for solving this problem was proposed. The proposed algorithm displayed promising performance for handling the trade-off between exploration and exploitation in the formulated problem. In Paper III, the main research outcomes include the systematic study of on- and off-policy policy optimization algorithms for SMILES-based *de novo* drug design. This also includes the comparison of different ways to learn from both current and previously generated samples. To generate a structurally diverse set of molecules, it is crucial to penalize the generation of similar structures. To facilitate agents not utilizing a copy of the pre-trained policy, it could be interesting to consider gradual penalization of similar molecules, e.g., based on the number of similar molecules, compared to the step function penalization used in this work. It can also be essential to distinguish between generated SMILES that are chemically invalid and molecules being penalized for being similar to previously generated molecules.

4.1 Future Directions

The research area of reinforcement learning is constantly making progress with several different research directions. This enables interesting future directions in utilizing reinforcement learning for *de novo* drug design. Recent work in *de novo* drug design has focused on sampling efficiency, for which reinforcement learning has demonstrated promising performance (Gao et al., 2022; Thomas, O'Boyle et al., 2022). Hence, it would be interesting to further investigate how to improve the sample efficiency in deep reinforcement learning for *de novo* drug design. For this purpose, different sampling strategies could be studied. This work only considers multimodal sampling without temperature, wherein future work can compare temperature sampling and beam search. A future direction for investigating the sample efficiency can also be to analyze the convergence rate to the optimal policy of current deep learning-based algorithms. For instance, the neural tangent kernel provides a technique to analyze the convergence in neural networks (Jacot, Gabriel & Hongler, 2018). To improve sample efficiency, and because experimental data is both costly and time-consuming to obtain, a future direction could be to further investigate offline reinforcement learning, which learns from stored data (Levine et al., 2020).

Another interesting problem is the design of the scoring function (i.e., objective function), which consists of several scoring components. Inverse reinforcement learning considers the problem of learning an agent's objectives, e.g., by observing a human expert (Arora & Doshi, 2021). Also, instead of only having one agent trying to maximize a reward function of several components, one could investigate the use of multiple agents where each agent tries to optimize a specific reward component while globally maximizing the cumulative reward by sharing knowledge.

Bibliography

Abe, N., Biermann, A. W., & Long, P. M. (2003). Reinforcement learning with immediate rewards and linear hypotheses. *Algorithmica*, *37*(4), 263–293 (cit. on p. 15).

Agrawal, R., Hegde, M., Teneketzis, D., et al. (1990). Multi-armed bandit problems with multiple plays and switching cost. *Stochastics and Stochastic reports*, *29*(4), 437–459 (cit. on p. 15).

Agrawal, S., & Goyal, N. (2013). Thompson sampling for contextual bandits with linear payoffs. *International conference on machine learning*, 127–135 (cit. on p. 15).

Åkerblom, N., Chen, Y., & Chehreghani, M. H. (2020). An online learning framework for energy-efficient navigation of electric vehicles. In C. Bessiere (Ed.), *Proceedings of the twenty-ninth international joint conference on artificial intelligence, IJCAI* (pp. 2051–2057). (Cit. on p. 14).

Åkerblom, N., Hoseini, F. S., & Chehreghani, M. H. (2022). Online learning of network bottlenecks via minimax paths. *Mach. Learn.* https://doi.org/10.1007/s10994-022-06270-0 (cit. on p. 14)

Anantharam, V., Varaiya, P., & Walrand, J. (1987). Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays-part i: Iid rewards. *IEEE Transactions on Automatic Control*, *32*(11), 968–976 (cit. on p. 15).

Anthony, T., Tian, Z., & Barber, D. (2017). Thinking fast and slow with deep learning and tree search. (Cit. on p. 18).

Arora, S., & Doshi, P. (2021). A survey of inverse reinforcement learning: Challenges, methods and progress. *Artificial Intelligence*, *297*, 103500 (cit. on p. 26).

Arulkumaran, K., Deisenroth, M. P., Brundage, M., & Bharath, A. A. (2017). Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, *34*(6), 26–38 (cit. on p. 18).

Arús-Pous, J., Johansson, S. V., Prykhodko, O., Bjerrum, E. J., Tyrchan, C., Reymond, J.-L., Chen, H., & Engkvist, O. (2019). Randomized smiles strings improve the quality of molecular generative models. *Journal of cheminformatics*, *11*(1), 1–13 (cit. on p. 8).

Auer, P. (2002). Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, *3*(Nov), 397–422 (cit. on p. 15).

Auer, P., Cesa-Bianchi, N., & Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine learning*, *47*(2), 235–256 (cit. on pp. 15, 16).

Bemis, G. W., & Murcko, M. A. (1996). The properties of known drugs. 1. molecular frameworks. *Journal of medicinal chemistry*, *39*(15), 2887–2893 (cit. on p. 9).

Bender, A., & Glen, R. C. (2004). Molecular similarity: A key technique in molecular informatics. *Organic & biomolecular chemistry*, *2*(22), 3204–3218 (cit. on p. 10).

Bilodeau, C., Jin, W., Jaakkola, T., Barzilay, R., & Jensen, K. F. (2022). Generative models for molecular discovery: Recent advances and challenges. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, *12*(5), e1608 (cit. on pp. 3, 4).

Bnaya, Z., Puzis, R., Stern, R., & Felner, A. (2013). Volatile multi-armed bandits for guaranteed targeted social crawling. *AAAI (Late-Breaking Developments)*, *2*(2.3), 16–21 (cit. on p. 15).

Bossér, J. D., Sörstadius, E., & Chehreghani, M. H. (2021). Model-centric and data-centric aspects of active learning for deep neural networks. *2021 IEEE International Conference on Big Data (Big Data)*, 5053–5062 (cit. on p. 19).

Bouneffouf, D., Rish, I., & Aggarwal, C. (2020). Survey on applications of multi-armed and contextual bandits. *2020 IEEE Congress on Evolutionary Computation (CEC)*, 1–8 (cit. on p. 15).

Brown, N., Fiscato, M., Segler, M. H., & Vaucher, A. C. (2019). Guacamol: Benchmarking models for de novo molecular design. *Journal of chemical information and modeling*, *59*(3), 1096–1108 (cit. on pp. 19, 23).

Chu, W., Li, L., Reyzin, L., & Schapire, R. (2011). Contextual bandits with linear payoff functions. *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 208–214 (cit. on p. 15).

Chua, K., Calandra, R., McAllister, R., & Levine, S. (2018). Deep reinforcement learning in a handful of trials using probabilistic dynamics models. (Cit. on p. 18).

Coley, C. W., Green, W. H., & Jensen, K. F. (2018). Machine learning in computer-aided synthesis planning. *Accounts of chemical research*, *51*(5), 1281–1289 (cit. on p. 3).

Danishuddin & Khan, A. U. (2016). Descriptors and their selection methods in qsar analysis: Paradigm for drug design. *Drug Discovery Today*, *21*(8), 1291–1302. https://doi.org/https://doi.org/10.1016/j.drudis.2016.06.013 (cit. on p. 10)

David, L., Thakkar, A., Mercado, R., & Engkvist, O. (2020). Molecular representations in ai-driven drug discovery: A review and practical guide. *Journal of Cheminformatics*, *12*(1), 1–22 (cit. on pp. 7, 8).

Eyke, N. S., Green, W. H., & Jensen, K. F. (2020). Iterative experimental design based on active machine learning reduces the experimental burden associated with reaction screening. *Reaction Chemistry & Engineering*, *5*(10), 1963–1972 (cit. on p. 19).

Gao, W., Fu, T., Sun, J., & Coley, C. (2022). Sample efficiency matters: A benchmark for practical molecular optimization. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho & A. Oh (Eds.), *Advances in neural information processing systems* (pp. 21342–21357). Curran Associates, Inc. (Cit. on pp. 11, 23, 26).

Gaulton, A., Bellis, L. J., Bento, A. P., Chambers, J., Davies, M., Hersey, A., Light, Y., McGlinchey, S., Michalovich, D., Al-Lazikani, B., et al. (2012). Chembl: A large-scale bioactivity database for drug discovery. *Nucleic acids research*, *40*(D1), D1100–D1107 (cit. on p. 12).

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., et al. (2018). Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905* (cit. on p. 18).

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, *2*(5), 359–366 (cit. on p. 18).

Hu, Y., Stumpfe, D., & Bajorath, J. (2016). Computational exploration of molecular scaffolds in medicinal chemistry: Miniperspective. *Journal of medicinal chemistry*, *59*(9), 4062–4076. https://doi.org/10.1021/acs.jmedchem.6b01437 (cit. on p. 9)

Hughes, J. P., Rees, S., Kalindjian, S. B., & Philpott, K. L. (2011). Principles of early drug discovery. *British journal of pharmacology*, *162*(6), 1239–1249 (cit. on pp. 3, 5).

Jacot, A., Gabriel, F., & Hongler, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi & R. Garnett (Eds.), *Advances in neural information processing systems*. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2018/file/5a4be1fa34e62bb8a6ec6b91d2462f5a-Paper.pdf. (Cit. on p. 26)

Johansson, S., Thakkar, A., Kogej, T., Bjerrum, E., Genheden, S., Bastys, T., Kannas, C., Schliep, A., Chen, H., & Engkvist, O. (2019). Ai-assisted synthesis prediction. *Drug Discovery Today: Technologies*, *32*, 65–72 (cit. on p. 11).

Kaiser, L., Babaeizadeh, M., Milos, P., Osinski, B., Campbell, R. H., Czechowski, K., Erhan, D., Finn, C., Kozakowski, P., Levine, S., Mohiuddin, A., Sepassi, R., Tucker, G., & Michalewski, H. (2020). Model-based reinforcement learning for atari. (Cit. on p. 18).

Kleinberg, R., Niculescu-Mizil, A., & Sharma, Y. (2010). Regret bounds for sleeping experts and bandits. *Machine learning*, *80*(2), 245–272 (cit. on p. 15).

Kleinberg, R., Slivkins, A., & Upfal, E. (2008). Multi-armed bandits in metric spaces. *arXiv preprint arXiv:0809.4882*. https://doi.org/10.48550/arXiv.0809.4882 (cit. on p. 16)

Kleinberg, R., Slivkins, A., & Upfal, E. (2019). Bandits and experts in metric spaces. *Journal of the ACM (JACM)*, *66*(4), 1–77 (cit. on p. 16).

Komiyama, J., Honda, J., & Nakagawa, H. (2015). Optimal regret analysis of thompson sampling in stochastic multi-armed bandit problem with

multiple plays. In F. Bach & D. Blei (Eds.), *Proceedings of the 32nd international conference on machine learning* (pp. 1152–1161). PMLR. (Cit. on p. 15).

Lai, T. L., Robbins, H., et al. (1985). Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, *6*(1), 4–22 (cit. on p. 13).

Lattimore, T., & Szepesvári, C. (2020). *Bandit algorithms*. Cambridge University Press. (Cit. on p. 14).

Levine, S., Kumar, A., Tucker, G., & Fu, J. (2020). Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643* (cit. on p. 26).

Li, L., Chu, W., Langford, J., & Schapire, R. E. (2010a). A contextual-bandit approach to personalized news article recommendation. *Proceedings of the 19th International Conference on World Wide Web, WWW*, 661–670 (cit. on p. 14).

Li, L., Chu, W., Langford, J., & Schapire, R. E. (2010b). A contextual-bandit approach to personalized news article recommendation. *Proceedings of the 19th international conference on World wide web*, 661–670 (cit. on p. 15).

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2015). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (cit. on p. 18).

Macarron, R., Banks, M. N., Bojanic, D., Burns, D. J., Cirovic, D. A., Garyantes, T., Green, D. V., Hertzberg, R. P., Janzen, W. P., Paslay, J. W., et al. (2011). Impact of high-throughput screening in biomedical research. *Nature reviews Drug discovery*, *10*(3), 188–195 (cit. on p. 5).

Martin, Y. C. (2009). Let's not forget tautomers. *Journal of computer-aided molecular design*, *23*, 693–704 (cit. on p. 12).

Masand, V. H., Mahajan, D. T., Ben Hadda, T., Jawarkar, R. D., Alafeefy, A. M., Rastija, V., & Ali, M. A. (2014). Does tautomerism influence the outcome of qsar modeling? *Medicinal Chemistry Research*, *23*, 1742–1757 (cit. on p. 12).

Mauri, A., Consonni, V., & Todeschini, R. (2017). Molecular descriptors. In *Handbook of computational chemistry* (pp. 2065–2093). Springer. (Cit. on p. 8).

Mennen, S. M., Alhambra, C., Allen, C. L., Barberis, M., Berritt, S., Brandt, T. A., Campbell, A. D., Castañón, J., Cherney, A. H., Christensen, M., et al. (2019). The evolution of high-throughput experimentation in pharmaceutical development and perspectives on the future. *Organic Process Research & Development*, *23*(6), 1213–1242 (cit. on p. 7).

Meyers, J., Fabian, B., & Brown, N. (2021). De novo molecular design and generative models. *Drug Discovery Today*, *26*(11), 2707–2715 (cit. on pp. 11, 19).

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., & Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In M. F. Balcan & K. Q. Weinberger (Eds.), *Proceedings of the 33rd international conference on machine learning* (pp. 1928–

1937). PMLR. https://proceedings.mlr.press/v48/mniha16.html. (Cit. on p. 18)

Morgan, H. L. (1965). The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service. *Journal of chemical documentation*, *5*(2), 107–113 (cit. on p. 8).

Muratov, E. N., Bajorath, J., Sheridan, R. P., Tetko, I. V., Filimonov, D., Poroikov, V., Oprea, T. I., Baskin, I. I., Varnek, A., Roitberg, A., et al. (2020). Qsar without borders. *Chemical Society Reviews*, *49*(11), 3525–3564 (cit. on p. 10).

Neil, D., Segler, M., Guasch, L., Ahmed, M., Plumbley, D., Sellwood, M., & Brown, N. (2018). Exploring deep recurrent models with reinforcement learning for molecule design [In: 6th International Conference on Learning Representations]. (Cit. on pp. 19, 23).

Neves, B. J., Braga, R. C., Melo-Filho, C. C., Moreira-Filho, J. T., Muratov, E. N., & Andrade, C. H. (2018). Qsar-based virtual screening: Advances and applications in drug discovery. *Frontiers in pharmacology*, *9*, 1275 (cit. on p. 10).

Paul, S. M., Mytelka, D. S., Dunwiddie, C. T., Persinger, C. C., Munos, B. H., Lindborg, S. R., & Schacht, A. L. (2010). How to improve r&d productivity: The pharmaceutical industry's grand challenge. *Nature reviews Drug discovery*, *9*(3), 203–214 (cit. on p. 3).

Polishchuk, P. G., Madzhidov, T. I., & Varnek, A. (2013). Estimation of the size of drug-like chemical space based on gdb-17 data. *Journal of computer-aided molecular design*, *27*, 675–679 (cit. on p. 3).

Press, W. H. (2009). Bandit solutions provide unified ethical models for randomized clinical trials and comparative effectiveness research. *Proc. Natl. Acad. Sci. USA*, *106*(52), 22387–22392 (cit. on p. 14).

Reymond, J.-L., & Awale, M. (2012). Exploring chemical space for drug discovery using the chemical universe database. *ACS chemical neuroscience*, *3*(9), 649–657 (cit. on p. 3).

Robbins, H. (1952). Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, *58*(5), 527–535 (cit. on p. 13).

Rogers, D., & Hahn, M. (2010). Extended-connectivity fingerprints. *Journal of chemical information and modeling*, *50*(5), 742–754 (cit. on p. 8).

Saikin, S. K., Kreisbeck, C., Sheberla, D., Becker, J. S., & Aspuru-Guzik, A. (2019). Closed-loop discovery platform integration is needed for artificial intelligence to make an impact in drug discovery. *Expert opinion on drug discovery*, *14*(1), 1–4 (cit. on p. 4).

Sanchez-Lengeling, B., & Aspuru-Guzik, A. (2018). Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, *361*(6400), 360–365 (cit. on p. 11).

Schaul, T., Quan, J., Antonoglou, I., & Silver, D. (2015). Prioritized experience replay. *arXiv preprint arXiv:1511.05952* (cit. on p. 18).

Schein, A. I., & Ungar, L. H. (2007). Active learning for logistic regression: An evaluation (cit. on p. 13).

Schneider, G. (2018). Automating drug discovery. *Nature reviews drug discovery*, *17*(2), 97–113 (cit. on p. 3).

Schneider, P., Walters, W. P., Plowright, A. T., Sieroka, N., Listgarten, J., Goodnow Jr, R. A., Fisher, J., Jansen, J. M., Duca, J. S., Rush, T. S., et al. (2020). Rethinking drug design in the artificial intelligence era. *Nature Reviews Drug Discovery*, *19*(5), 353–364 (cit. on p. 10).

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (cit. on p. 18).

Schwaller, P., Laino, T., Gaudin, T., Bolgar, P., Hunter, C. A., Bekas, C., & Lee, A. A. (2019). Molecular transformer: A model for uncertainty-calibrated chemical reaction prediction. *ACS central science*, *5*(9), 1572–1583 (cit. on p. 11).

Schwaller, P., Vaucher, A. C., Laino, T., & Reymond, J.-L. (2021). Prediction of chemical reaction yields using deep learning. *Machine learning: science and technology*, *2*(1), 015016 (cit. on p. 11).

Segler, M. H., & Waller, M. P. (2017). Neural-symbolic machine learning for retrosynthesis and reaction prediction. *Chemistry–A European Journal*, *23*(25), 5966–5971 (cit. on p. 11).

Settles, B. (2012). *Active learning*. Morgan & Clay Pool. https://doi.org/10.2200/S00429ED1V01Y201207AIM018. (Cit. on p. 12)

Shen, W., Wang, J., Jiang, Y., & Zha, H. (2015). Portfolio choices with orthogonal bandit learning. In Q. Yang & M. J. Wooldridge (Eds.), *Proceedings of the twenty-fourth international joint conference on artificial intelligence, IJCAI* (p. 974). (Cit. on p. 14).

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K., & Hassabis, D. (2017). Mastering chess and shogi by self-play with a general reinforcement learning algorithm. (Cit. on p. 18).

Slivkins, A. (2011). Contextual bandits with similarity information. In S. M. Kakade & U. von Luxburg (Eds.), *Proceedings of the 24th annual conference on learning theory* (pp. 679–702). PMLR. (Cit. on pp. 16, 20, 22).

Slivkins, A. (2022). Introduction to multi-armed bandits. *arXiv*. https://doi.org/10.48550/arXiv.1904.07272 (cit. on pp. 14, 16)

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press. (Cit. on p. 17).

Szepesvári, C. (2010). Algorithms for reinforcement learning. *Synthesis lectures on artificial intelligence and machine learning*, (9), Algorithms for reinforcement learning. https://doi.org/10.2200/S00268ED1V01Y201005AIM009 (cit. on p. 17)

Thomas, M., O'Boyle, N. M., Bender, A., & De Graaf, C. (2022). Augmented hill-climb increases reinforcement learning efficiency for language-based de novo molecule generation. *Journal of Cheminformatics*, *14*(1), 1–22 (cit. on pp. 19, 23).

Thomas, M., O'Boyle, N. M., Bender, A., & De Graaf, C. (2022). Re-evaluating sample efficiency in de novo molecule generation. *arXiv preprint arXiv:2212.01385* (cit. on pp. 11, 23, 26).

Tropsha, A. (2010). Best practices for qsar model development, validation, and exploitation. *Molecular informatics*, *29*(6-7), 476–488 (cit. on pp. 10, 12).

Tyrchan, C., Nittinger, E., Gogishvili, D., Patronov, A., & Kogej, T. (2022). Chapter 4 - approaches using ai in medicinal chemistry. In T. Akitsu (Ed.), *Computational and data-driven chemistry using artificial intelligence* (pp. 111–159). Elsevier. https://doi.org/https://doi.org/10.1016/B978-0-12-822249-2.00002-5. (Cit. on pp. 10, 11)

Vamathevan, J., Clark, D., Czodrowski, P., Dunham, I., Ferran, E., Lee, G., Li, B., Madabhushi, A., Shah, P., Spitzer, M., et al. (2019). Applications of machine learning in drug discovery and development. *Nature reviews Drug discovery*, *18*(6), 463–477 (cit. on p. 3).

Villar, S. S., Bowden, J., & Wason, J. (2015). Multi-armed bandit models for the optimal design of clinical trials: Benefits and challenges. *Statistical science: a review journal of the Institute of Mathematical Statistics*, *30*(2), 199 (cit. on p. 14).

Vohora, D., & Singh, G. (2017). *Pharmaceutical medicine and translational clinical research*. Academic Press. (Cit. on p. 6).

Wang, Z., Bapst, V., Heess, N., Mnih, V., Munos, R., Kavukcuoglu, K., & de Freitas, N. (2016). Sample efficient actor-critic with experience replay. *arXiv preprint arXiv:1611.01224* (cit. on p. 18).

Weininger, D. (1988). Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, *28*(1), 31–36 (cit. on p. 7).

Wildey, M. J., Haunso, A., Tudor, M., Webb, M., & Connick, J. H. (2017). Chapter five - high-throughput screening. In R. A. Goodnow (Ed.), *Platform technologies in drug discovery and validation* (pp. 149–195). Academic Press. https://doi.org/https://doi.org/10.1016/bs.armc.2017.08.004. (Cit. on p. 5)

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Reinforcement learning*, 5–32 (cit. on p. 18).

Wouters, O. J., McKee, M., & Luyten, J. (2020). Estimated research and development investment needed to bring a new medicine to market, 2009-2018. *Jama*, *323*(9), 844–853 (cit. on p. 3).

Yang, Y., Ma, Z., Nie, F., Chang, X., & Hauptmann, A. G. (2015). Multi-class active learning by uncertainty sampling with diversity maximization. *International Journal of Computer Vision*, *113*, 113–127 (cit. on p. 13).

Young, D., Martin, T., Venkatapathy, R., & Harten, P. (2008). Are the chemical structures in your qsar correct? *QSAR & combinatorial science*, *27*(11-12), 1337–1345 (cit. on p. 10).

Zheng, S., Rao, J., Zhang, Z., Xu, J., & Yang, Y. (2019). Predicting retrosynthetic reactions using self-corrected transformer neural networks.

Journal of chemical information and modeling, *60*(1), 47–55 (cit. on p. 11).