Democratic and Popular Republic of Algeria
الجمهورية الجزائرية الديمقراطية الشعبية
Ministry of Higher Education and Scientific Research
وزارة التعليم العالي و البحث العلمي

جامعة يحيى فارس المدية
Yahia Fares University of Medea
Faculty of science
Department of mathematics and computer science



Thesis

Submitted in Partial Fulfillment of the Requirements for Degree of Master in Computer Science

Speciality: Systems engineering and web technologies

Ligand Based Virtual screening using Reinforcement Learning

Presented by :
Mr. Abdelmadjid Benimam
Mr. Riadh Marmouze

Supervisor:
Mr. Hamza Hentabli
Co-supervisor:
Mr. Brahim Sahmadi

Defended on June 15, 2023, In front of the jury composed of:

Mr. Redha Mazari :	Yahia Fares University	- President
Mrs. Nadjah Chergui :	Yahia Fares University	- Examiner

Class of : 2022/2023

Abstract

Ligand-based virtual screening (LBVS) is a widely employed strategy in drug discovery to identify potential ligands for target proteins based on molecular structure similarity. However, conventional LBVS methods often rely on manually designed molecular descriptors or fingerprints that may not capture the essential characteristics necessary for binding affinity and specificity. This thesis proposes a novel LBVS methodology utilizing a computer-learned representation of molecules derived from features extracted through reinforcement learning (RL) with the Proximal Policy Optimization (PPO) algorithm. The RL agent is trained to perform multi-label classification on a diverse and extensive dataset of drug-like molecules (MDDR), employing a custom-designed environment that facilitates episodic data access. Moreover, a meticulously designed reward function is implemented to incentivize the agent for accurate classification. The acquired representation enables similarity computation and ranking of query and database molecules. Our method is evaluated on 10 reference structures from diverse protein families and compared against state-of-the-art LBVS techniques. The findings demonstrate significant enhancements in the top 1% and 5% cut-off recall values, with an average improvement of 50.6% at the 1% cut-off and 20.73% at the 5% cut-off, effectively showcasing the efficacy of our proposed approach.

Keywords : Ligand-based virtual screening (LBVS), Drug discovery, similarity searching, Molecular descriptors, Reinforcement learning (RL), Proximal Policy Optimization (PPO), Multi-label classification.

ملخص

يعد الفحص الافتراضي المستند إلى الربائط (LBVS) استراتيجية مستخدمة على نطاق واسع في اكتشاف الأدوية لتحديد الروابط المحتملة للبروتينات المستهدفة بناءً على تشابه البنية الجزيئية. ومع ذلك ، غالبًا ما تعتمد طرق التقليدية على الواصفات الجزيئية المصممة يدويًا أو البصمات التي قد لا تلتقط الخصائص الأساسية اللازمة للارتباط والخصوصية. تقترح هذه الأطروحة منهجية LBVS جديدة تعتمد على التمثيل الحاسوبي للجزيئات المشتقة من الميزات المستخرجة من خلال التعلم المعزز (RL) باستخدام خوارزمية تحسين السياسة(الاستراتيجية) القريبة (PPO). تم تدريب وكيل على إجراء تصنيف متعدد العلامات على مجموعة بيانات متنوعة وشاملة من الجزيئات الشبيهة بالعقاقير (MDDR) ، باستخدام بيئة مصممة خصيصًا تسهل الوصول إلى البيانات بشكل عرضي . علاوة على ذلك ، تم تصميم دالة مكافأة جديدة بدقة لتحفيز الوكيل على التصنيف الدقيق. يتيح هذا التمثيل المكتسب تسهيل حساب التشابه وترتيب الجزيئات المرجعية وقاعدة البيانات. تم تقييم طريقتنا على 10 هياكل مرجعية من عائلات بروتين متنوعة ومقارنتها بأحدث تقنيات LBVS .أظهرت النتائج تحسينات كبيرة لقيم الإستذكار في أعلى قيم القطع 1% و 5% ، مع تحسن متوسط بنسبة 50.6% عند الحد الفاصل 1% و 20.73% عند الحد الفاصل 5% ، مما يُظهر فعالية نهجنا المقترح.

كلمات مفتاحية : الفحص الإفتراضي القائم على الربائط، إكتشاف العقار، البحث عن التشابه، الواصفات الجزيئية، التعلم المعزز (RL)، تحسين السياسة القريبة (PPO)، التصنيف متعدد العلامات.

Acknowledgment

We would like to express our sincere gratitude to H.Hentabli, who played an invaluable role in the successful completion of our master's thesis. H. Hentabli generously provided us with the dataset essential for our research and actively contributed to the execution of our research activities. Their guidance, support, and expertise greatly enhanced the quality and depth of our study.

We are also grateful to B.Sahmadi, who served as our co-supervisor and provided valuable insights throughout the entire thesis process. Their assistance in proofreading and refining our work was instrumental in ensuring the clarity and coherence of our findings. Their attention to detail and constructive feedback significantly improved the overall quality of our thesis.

Additionally, we would like to extend our thanks to all the individuals who participated in this research study, as well as those who provided us with valuable advice, resources, and encouragement along the way.

Lastly, we express our heartfelt appreciation to our families, friends, and colleagues who supported us during the challenging moments of this thesis journey. Their unwavering encouragement and belief in our abilities were a constant source of motivation.

We are truly grateful for the contributions and support from all those involved, without whom this thesis would not have been possible.

Abbreviations

CNN	Convolutional neural network
DQN	Deep Q-Network
DRL	Deep Reinforcement Learning
LBVS	Ligand-based virtual screening
MDDR	MLD Drug Dataset Report
MDP	Markov Decision Process
MHA	Multi-head attention
MLP	Multi-Layer Perceptron
PPO	Proximal policy optimization
QSPR	Quantitative structure–property relationship
QSAR	Quantitative structure–activity relationship
RL	Reinforcement learning
SL	Supervised Learning
SMILES	Simplified Molecular-Input Line-Entry System
TRPO	Trust region policy optimization
SBVS	Structure-based virtual screening
UL	Unsupervised Learning
VS	Virtual Screening

Contents

Abstract ii

Acknowledgment iv

Abbreviations v

Contents vi

List of Figures ix

List of Tables xi

1 Introduction 1

1.1 Problem Statement . . . 3
1.2 Research Questions . . . 3
1.3 Research Objectives . . . 3
1.4 Research Scope . . . 4
1.5 Thesis Organization . . . 4

2 Literature Review 5

2.1 Introduction . . . 5
2.1.1 Chemoinformatics Definition . . . 6
2.1.2 Chemoinformatics in drug discovery . . . 7
2.2 Virtual screening . . . 8
2.2.1 Structure-based Virtual screening (SBVS) . . . 8
2.2.2 Ligand-based Virtual screening (LBVS) . . . 9
2.2.3 Limitations of Virtual Screening . . . 10
2.3 Ligand-based Virtual Screening methods . . . 11
2.3.1 Similarity searching algorithms . . . 11

2.3.2 Pharmacophore modeling methods . . . 12
2.4 Chemical descriptors . . . 13
2.4.1 String/Line-Based Representations . . . 14
2.4.2 Chemical Table-Based Representations . . . 15
2.5 Feature-Based Representations . . . 15
2.5.1 Extended Connectivity Fingerprints (ECFP) . . . 16
2.6 Molecular Similarity Measurement . . . 18
2.6.1 Structural Similarity . . . 18
2.7 Machine Learning . . . 20
2.7.1 Supervised Learning . . . 20
2.7.2 Unsupervised Learning . . . 21
2.7.3 Reinforcement learning . . . 21
2.8 Convolutional Neural Network (CNN) architecture . . . 28
2.9 Multi-Head attention architecture . . . 29
2.10 Related work . . . 31
2.10.1 Bayesian inference network (BIN) in LBVS . . . 31
2.10.2 Siamese Networks for LBVS . . . 33
2.10.3 Convolutional Neural Networks for LBVS . . . 35
2.10.4 Stacked deep belief network (SDBN) in LBVS . . . 35
2.11 Conclusion . . . 36

3 Research Methodology 38
3.1 Introduction . . . 38
3.2 Research Design . . . 38
3.2.1 Research Framework . . . 38
3.2.2 Datasets . . . 41
3.3 Design and Steps . . . 43
3.3.1 Data preprocessing . . . 43
3.3.2 Merging Dataset . . . 44
3.3.3 Framework Implementation and network architecture . . . 46
3.3.4 Proximal policy optimization agent architecture . . . 48
3.4 Benchmark : Ligand-based virtual screening process . . . 58
3.4.1 LBVS design . . . 58
3.4.2 LBVS performance metrics . . . 59
3.5 Evaluation metrics . . . 60
3.5.1 Jaccard Score . . . 60
3.5.2 Hamming loss . . . 60

3.5.3 Recall, Precision, F-measure 61
3.6 Concluding Summary 62

4 Results and Discussion **63**
4.1 Introduction 63
4.2 Implementation Tools 63
4.3 Results 64
4.3.1 Classification Model Results 65
4.3.2 Training results 65
4.3.3 Test results 67
4.3.4 LBVS Results 70
4.4 Discussion 80
4.4.1 Implications of the results 80
4.4.2 Improvements to the architecture 81
4.4.3 Improvements to the dataset 81

5 Conclusion **82**
5.1 Research contribution 82
5.2 Future work 83
5.3 Summary 84

Bibliography **85**

List of Figures

2.1 Graphical representation of cheminformatics strategies. 7
2.2 Overview of the different classes of molecular representations. 13
2.3 Schematic diagram of Morgan fingerprint generation. 16
2.4 ECFP generation process 17
2.5 Generation of binary vector representation using calculated identifiers. . . 17
2.6 Value function illustration. 25
2.7 A Convolutional Neural Network (CNN) architecture for image classification. 28
2.8 Overview of multi-head attention architecture. 30

3.1 Flow Chart illustrating Research Framework's phases 39
3.2 Steps of converting molecule data to bit-string fingerprint. 44
3.3 A sample of created dataset. 45
3.4 DS1 activity class depicting indexes of biological activities 45
3.5 Distribution of compounds across the classes of newly Merges Dataset . . . 46
3.6 Overall design of the Multi-label custom environment. 47
3.7 A diagram illustrating how PPO algorithm optimizes a stochastic policy. 49
3.8 Overall architecture of PPO network showing main three-part of the network (Feature extractor, mlp extractor, action/value network) 50
3.9 Overall architecture of Feature extractor 51
3.10 Detailed architecture of CNN module in feature extractor 52
3.11 Detailed architecture of linear module in feature extractor 53
3.12 Detailed architecture of MHA (Multi-Head Attention) module in feature extractor 54
3.13 Detailed architecture of linear module in mlp extractor 56
3.14 Detailed architecture of action predictor network 57
3.15 Detailed architecture of value predictor network 57
3.16 Overall process of ligand-based virtual screening 58

4.1 Performance of an agent trained for 7 million timesteps. The top left plot shows the average episode reward during training, while the top right plot shows the training loss over time. The bottom left plot shows the policy gradient loss over time, and the bottom right plot shows the value loss over time. 67
4.2 Classification test report 69
4.3 Comparison of the performance values (recall) of similarity methods for DS1 dataset in Top 1% 73
4.4 Comparison of the performance values (recall) of similarity methods for DS1 dataset in Top 5%. 74
4.5 Comparison of the performance values (recall) of similarity methods for DS2 dataset in Top 1%. 76
4.6 Comparison of the performance values (recall) of similarity methods for DS2 dataset in Top 5% 77
4.7 Comparison of the performance values (recall) of similarity methods for DS3 dataset in Top 1%. 79
4.8 Comparison of the performance values (recall) of similarity methods for DS3 dataset in Top 5%. 80

List of Tables

2.1 Classification of VS methods based on the amount of information available. 11
2.2 Common molecular similarity measures with their formulas. 19

3.1 Activity classes, pair-wise similarity, indexes and number of molecules for DS1. 42
3.2 Activity classes, pair-wise similarity, indexes and number of molecules for DS2. 42
3.3 Activity classes, pair-wise similarity, indexes and number of molecules for DS3. 43

4.1 Data distribution of classes between train and test 65
4.2 Used PPO parameters in stable-baselines3 library 66
4.3 The obtained results of recall values in top 1% for the DS1 dataset. 72
4.4 The obtained results of recall values in top 5% for the DS1 dataset. 72
4.5 The obtained results of recall values in top 1% for the DS2 dataset. 75
4.6 The obtained results of recall values in top 5% for the DS2 dataset. 75
4.7 The obtained results of recall values in top 1% for the DS3 dataset. 78
4.8 The obtained results of recall values in top 5% for the DS3 dataset. 78

Chapter 1

Introduction

Drug design is the process consisting of discovering and developing new drugs or refining existing ones to treat diseases based on knowledge of the biological target [1]. In general, it involves the identification and optimization of compounds that can have therapeutic effects on the body. The main goal of the drug design process is to develop effective and safe therapeutic products against one or more pathological diseases.

This process of creating a new drug is usually divided into several steps. It begins with the discovery and development phase, in which several thousand potential target compounds are selected based on various criteria then experiments will be conducted on these compounds to gather crucial information such as absorption, best dosage, and side effects. The next phase is preclinical research. Researchers must find out any potential toxicity or side effects before testing the drug on the human species. Usually, this step is not applied largely. In contrast, this phase must answer all related questions about toxicity levels and dosing. After that, it comes another important phase which is Clinical research where experiments will be directly done inside real human beings to make sure that the drugs meet the required standards as well as the desired effects on humans. In the end, the drugs will be approved by a specialized committee or an institution to be used in health institutions.

Following these steps is, on the one hand, a time-consuming task and, on the other hand, quite expensive and makes failures highly intolerable. In fact, the industry of pharmaceutical products spends an enormous amount of resources to develop a product. The probability of a new drug making it to the market is about 13% [2]. Due to these constraints, various interdisciplinary domains were introduced into the process to reduce the costs and time required for the development of new medicines. one of the most used domains is chemoinformatics.

Combining the fields of computer science and chemistry, chemoinformatics as term appeared recently to describe a domain that organized and coordinate the application of

computers in chemistry. in the hope of making better decisions in lead drug identification as well as optimization.

The core topics of chemoinformatics include the retrieval of chemical compounds, techniques for the management of chemical structure database, virtual screening methods, structure propriety prediction methods, similarity and diversity analysis, and classification of compounds for high-throughput screening [3].

Virtual screening is a computational technique that uses computer programs to identify potential hits from virtual fragment libraries. Most widely used approaches in VS are Ligand-based methods and docking methods. One noticeable advantage of virtual screening is cost-effectiveness and fast process compared to more traditional methods such as high-throughput screening.

Based on data from previously established active compounds Virtual screening tries to predict and identify structurally varied compounds with similar biological activity, regardless of the technique used. It is mainly based on the similarity propriety principle [4] where it states that molecules that have similar chemical structures lean to have similar bioactivity function. the importance of similarity search applications has increased significantly recently. this is due to the important roles of LBVS in lead optimization phase of drug design.

To calculate similarity, LBVS uses molecular descriptors called fingerprints. A fingerprint is a mathematical representation of chemical compounds. It is often represented as an array of Boolean values which represents a set of structural fragments of a molecule. Similarity measures such as Tanimato and Cosine are used to calculate molecular resemblance therefore molecules will be ranked based on the results of this measure.

Reinforcement learning as a sub-domain of machine learning stands on three basic concepts agent, actions, and environment. Precisely an intelligent agent is supposed to take action in a predefined environment where the goal is to maximize the notion of cumulative reward [5]. Furthermore, To maximize this cumulative reward, an agent must develop a policy. This latter is a description of how an agent must behave given the current state and the goal. Often for practical purposes of reinforcement learning, A policy must be represented with a proximal function such as a deep neural network. This fusion is usually called deep reinforcement learning. The most popular DRL algorithm is Proximal Policy Optimization PPO [6]. PPO is a policy gradient algorithm that handles continuous states and action spaces. one of its key features is that it takes less time to train, has less variance, and gives better and more stable results than other algorithms.

1.1 Problem Statement

Although LBVS methods were used extensively in the lead compound development phase. Traditional molecular descriptors used in the similarity search process suffer from high rates of inaccuracy. This is due to the virtual screening performance of interaction FingerPrint overestimation (IFP) [7]. Furthermore, traditional feature-based similarity searching methods are even more biased and delusional, Which makes their results unreliable. Moreover, the introduction of deep learning methods has led to significant improvement in the similarity-searching process when it comes to homogeneous biological activities. But these methods failed to replicate these results in bio-activity classes that contain high-diversity (heterogeneous) molecular structures.

1.2 Research Questions

Our research aims to address the following key questions related to the application of Deep Reinforcement Learning (DRL) in LBVS:

- Can the performance of LBVS be enhanced by leveraging a new computer-learned representation based on extracted features of Deep reinforcement learning model?
- Can a DRL agent's policy effectively discriminate between active compounds and decoys, surpassing the performance of state-of-the-art methods?
- Can similarity searching performance of heterogeneous structures be improved using the new computer-learned molecular representation?

1.3 Research Objectives

Given that LBVS requires accurate discrimination between active and inactive compounds, the primary objective of this thesis is to develop a novel molecular representation using a popular Deep Reinforcement Learning algorithm capable of achieving this level of differentiation. To accomplish this objective, we will pursue the following key goals:

- Analyze, compare, and review state-of-the-art LBVS approaches to identify their strengths and limitations.
- Study and comprehend the concepts of deep reinforcement learning and explore their applications in the field of LBVS.

- Generate and experiment with a new molecular descriptor derived from the DRL agent's learned representation.
- Compare the performance of the newly generated molecular representation with state-of-the-art methods.

1.4 Research Scope

This thesis aims to use a reinforcement learning (RL) agent for ligand-based virtual screening (LBVS), which is a technique to find potential drugs based on their similarity to known active molecules. We use a proximal policy optimization (PPO) agent to learn features from molecular graphs and classify the MDDR dataset. We evaluate and compare the PPO agent with other methods for LBVS. We expect to show that the PPO agent is feasible, effective, novel, and interpretable for LBVS.

1.5 Thesis Organization

This thesis comprises five chapters that address the role of Ligand-Based Virtual Screening (LBVS) in drug design, specifically in lead compound identification and optimization. The first chapter provides an introduction to the research problem, research questions, objectives, and scope of the study. In addition, this chapter provides an overview of LBVS and its importance in drug design.

The second chapter presents a comprehensive literature review of state-of-the-art LBVS methods and models, as well as an overview of deep reinforcement learning and its application in similar or related work to LBVS.

The third chapter focuses on the dataset used for the training and benchmarking phases. It provides a detailed description of the dataset, highlights the methods used during the training process, and explains the architecture of the policy network. Furthermore, the chapter describes the whole process of ligand-based virtual screening, including the similarity measures used.

The fourth chapter presents the results of the DRL agent training process, as well as the results of applying LBVS to MDDR benchmarking dataset. The chapter also discusses these results and concludes on the effects of using DRL in LBVS.

The fifth and final chapter provides a general conclusion, summarizes the key findings, and presents perspectives and future directions for research based on the outcomes of the study.

Chapter 2

Literature Review

2.1 Introduction

Electronic databases have been the primary means for chemists to share the accumulation of information they generate. However, a proper name for the field dealing with this form of information storage and manipulation was absent. Initially, scientists called it "Chemical Information," which did not accurately differentiate between librarianship and computer methods development [8].

To highlight the importance of computers in chemical information processing, some scientists adopted the term "Computer Chemistry." Nevertheless, others found this confusing as "Computational Chemistry" already existed as a distinct field focused on theoretical quantum mechanical calculations.

In 1975, The Journal of Chemical Documentation changed its name to incorporate Computer Sciences, reflecting the changing times. In 1973, experts from various branches of chemistry gathered at the first NATO Advanced Study Institute Summer School in Noordwijkerhout (The Netherlands) [8]. They shared their expertise in creating computer methods that manage and make sense of chemical data. During this Summer School exercise, different groups worked on database creation, software development for molecular modeling, organic synthesis design, analysis regarding spectral data, and chemometrics.

All of these contributions helped to build a new discipline that impacted many areas of chemistry. Informatics has since become more broadly applied across many areas within chemistry, solving complex tasks by employing mathematics and providing insights into fields previously restricted by traditional limitations [8].

In 1998, J. Gasteiger and T. Engel at the University of Erlangen-Nuremberg in Germany coined the term "chemoinformatics" to describe the emerging field of computer-based chemical information management and analysis [9]. This innovative methodology

integrates chemistry, computer science, mathematics, and statistics to analyze chemical data, revolutionizing how scientists think about chemistry as a discipline.

2.1.1 Chemoinformatics Definition

Chemoinformatics has multiple definitions, but the most widely accepted one is by Greg Paris, who defines it as "a generic term encompassing the design, creation, organization, management, retrieval, analysis, dissemination, visualization, and use of chemical information." [10].

The fundamental operations of any chemical information system involve storing data and information, furthermore retrieving and searching their relationships. Chemoinformatics has significant potential in gathering large-scale chemical data from public sources and designing new materials with specific properties using prediction methods that rely on physical, biological, or molecular chemical characteristics, such as spectra simulation, reaction modeling, synthesis planning, and drug design processes [11].

Chemoinformatics applies primarily to small molecules and has significant applications in medical science fields to develop innovative drugs with higher efficacy rates. It also finds use in material science to design superior-quality materials. The exploration of chemoinformatics also benefits allied fields, including agrochemicals and biotechnology. While traditional computer storage systems primarily deal with textual data structures, Chemical Information Systems predominantly house structural models requiring differing algorithms to handle their inherent complexities.

Moreover, modern high-performance computing infrastructure offers enhanced speed-to-market capabilities, paving the way to revolutionize current practices significantly while still maintaining efficiency standards.

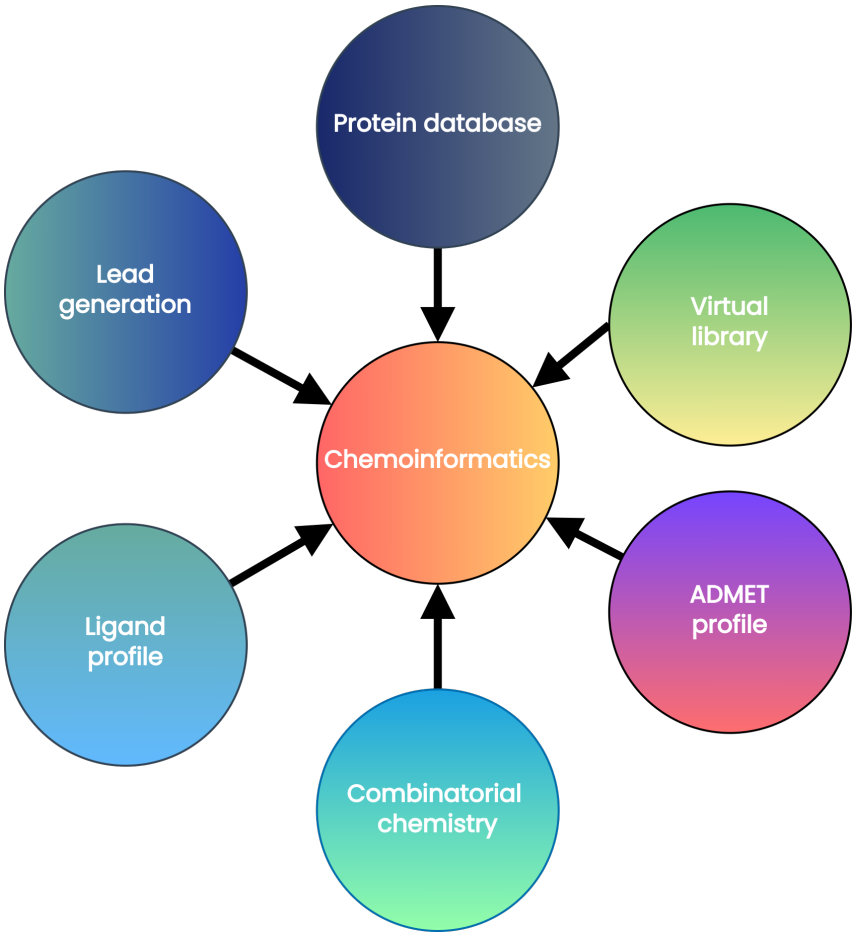


Figure 2.1: Graphical representation of cheminformatics strategies [12].

2.1.2 Chemoinformatics in drug discovery

Cheminformatics is a vast and multidimensional field that can be implemented in numerous areas. Its applications have diversified beyond medicinal chemistry to include biotechnology, biophysics, biochemistry, and related subjects [13]. One of its crucial aspects is its relevance to modern drug development.

Cheminformatics provides key tools for mainstream drug discovery, such as computer-aided drug synthesis, identification of pharmacophore moieties, and database development (Figure 2.1) [14]. Besides, it has several other applications, including data modeling, analysis, spectroscopy, environment-related investigations, and regulatory compliance checks. Toxicology studies also use chemoinformatics techniques.

Combinatorial chemistry has profoundly impacted computational chemistry and the refinement process in recent years. More precisely, computational methods can determine conformational characterizations and explore molecular interactions, enhancing molecular modeling approaches. The remarkable advancements in chemoinformatics have facilitated

the use of top chemical entities to target diseases more efficiently with less effort.

2.2 Virtual screening

The technique of virtual screening has gained significant popularity in drug discovery process. Not only is it a cheaper alternative to traditional High Throughput Screening (HTS) testing, but it can also identify false negatives that might be missed in other methods. However, some researchers have found that virtual screening may not be as effective as HTS when it comes to identifying compounds with unique structural diversity. Despite this limitation, there is evidence suggesting that virtual screening can still yield promising results by identifying potential drug candidates through simulations. In fact, numerous studies have highlighted the advantages of using computational approaches for drug discovery purposes.

As opposed to the HTS approach, which involves a haphazard exploration through a collection of compounds, the Virtual Screening (VS) technique is a more systematic and deliberate endeavor. In contrast to random searches characterized by HTS, VS methods rely on an advanced level of knowledge-driven analysis for effective results. This defining characteristic of VS has been widely recognized and emphasized in various classifications of screening methods, as depicted in (Table 2.1) (as modified from Bielska et al. [15]).

In order to conduct a structure-based virtual screening, it is necessary to have information on the known ligand(s) as well as the known structure of the target or a close homologue. There are two primary methods for conducting this type of virtual screening: protein-ligand docking and High Throughput Molecular Dynamics [16]. Protein-ligand docking involves predicting the binding affinity between a small molecule (ligand) and a protein by simulating their interaction [17]. On the other hand, High Throughput Molecular Dynamics uses computational simulations to predict how small molecules will interact with proteins at a much larger scale than traditional molecular dynamics simulations. Both methods have proven effective in identifying potential drug candidates for various diseases and conditions, and continue to be important tools in drug discovery research.

2.2.1 Structure-based Virtual screening (SBVS)

When it comes to the discovery of new drugs, there are various methods employed by scientists. One approach is structure-based virtual screening which involves using computer programs to identify potential drug candidates that could interact with a specific protein target [18]. The success of this method depends on two important factors: whether or not ligands (small molecules) that can bind to the target have been identified and whether the

structure of the target protein or a similar protein has been determined. In cases where both of these conditions are not met, de novo structure-based virtual screening can be used instead. This method involves carrying out docking simulations between potential ligands and the target protein followed by High Throughput Molecular Dynamics simulations to assess their stability and dynamics in order to predict their ability to effectively interact with the target. By using these advanced computational tools, scientists can greatly increase their chances of finding promising new compounds for drug development.

2.2.2 Ligand-based Virtual screening (LBVS)

Ligand-based virtual screening (LBVS) is a computational technique that aims to identify novel bioactive compounds by screening large molecular databases using the structural and physicochemical properties of known ligands [19]. LBVS can be applied when the target protein structure is unknown or unreliable, or when the binding site is flexible or allosteric [20]. LBVS methods can be broadly classified into two categories: similarity searching and pharmacophore modeling [19].

Similarity searching is based on the principle that structurally similar molecules tend to have similar biological activities. Similarity searching methods use molecular descriptors or fingerprints to represent the features of molecules and calculate their pairwise similarity using various metrics [19]. Similarity searching can be performed using online web servers that allow users to screen publicly accessible chemical databases in a reasonable time. Some examples of web servers for similarity searching are SEA (Similarity Ensemble Approach), ChemMapper, and Pharmit [19].

Pharmacophore modeling is based on the assumption that the biological activity of a molecule depends on its ability to form specific interactions with the target protein. Pharmacophore modeling methods use pharmacophore features, such as hydrogen bond donors, acceptors, aromatic rings, and hydrophobic groups, to represent the key elements of molecular recognition [19]. Pharmacophore models can be derived from a set of known active ligands or from a single reference ligand. Pharmacophore models can be used to screen molecular databases for compounds that match the pharmacophore features and constraints. Some examples of web servers for pharmacophore modeling are LigBuilder, PharmScreen, and PharmMapper [19].

LBVS methods have several advantages over structure-based virtual screening (SBVS) methods, such as being less computationally demanding, more tolerant to conformational changes, and more applicable to diverse targets. However, LBVS methods also have some limitations, such as being dependent on the quality and diversity of the training set, being unable to discover novel scaffolds, and being susceptible to false positives and

negatives. Therefore, LBVS methods are often combined with SBVS methods in a hybrid approach that exploits the complementary strengths of both techniques. For example, LBVS methods can be used to pre-filter a large chemical library before docking, or SBVS methods can be used to refine and rank the hits obtained from LBVS methods [20].

LBVS is a powerful and versatile tool for drug discovery that can complement and enhance SBVS methods. LBVS methods have been successfully applied to various targets and therapeutic areas, such as kinases, GPCRs, proteases, and anti-cancer agents [19, 20]. With the availability of online web servers and large molecular databases, LBVS methods are becoming more accessible and efficient for researchers. However, LBVS methods still face some challenges in dealing with the complexity and diversity of molecular interactions, the accuracy and validity of molecular descriptors and similarity metrics, and the integration and optimization of hybrid strategies [19, 20].

2.2.3 Limitations of Virtual Screening

When it comes to dealing with unknown ligands and an unknown target structure, virtual screening is not possible. In this scenario, researchers would need to rely on alternative approaches such as experimental testing or computational modeling based on existing data sources. While these methods can still yield valuable insights into potential drug candidates, they do require considerably more time and resources than virtual screening techniques. Ultimately, the choice of approach will depend on factors such as available resources and the specific goals of the research project at hand.

The use of structural information can prove to be helpful in implementing specific non-exclusive techniques for addressing virtual screening tasks. Typically, there are two main types of strategies utilized in the most common screening methods: structure-based VS (which is further divided into de novo ligand methods) and ligand-based VS. Many researchers classify their work into one of these three categories, namely Structure-based VS, Ligand-based VS and Combinatorial or structure-based de novo design. However, some procedures are significant enough to warrant being classified as a separate category even though they belong to one of these major categories. As a result, it is also usual for certain methods to be classified under any of these other two categories: Chemogenomics and Machine Learning. These additional categorizations help provide a more comprehensive understanding when trying to identify suitable approaches for conducting virtual screenings efficiently and effectively.

	Known ligand(s)	Unknown ligands
Known structure of target or close homologue	structure-based VS: - protein-ligand docking - High Throughput Molecular Dynamics	de novo structure-based VS: - protein-ligand docking - High Throughput Molecular Dynamics
Unknown target structure	Ligand-based VS: - few ligands: similarity search - several ligands: pharmacophore search, QSAR	virtual screening cannot be applied

Table 2.1: Classification of VS methods based on the amount of information available (adapted from the one given by Bielska et al [15].)

2.3 Ligand-based Virtual Screening methods

Ligand-based virtual screening (LBVS) is a computational technique that aims to identify novel bioactive compounds by screening large molecular databases using the structural and physicochemical properties of known ligands[21]. LBVS can be applied when the target protein structure is unknown or unreliable, or when the binding site is flexible or allosteric[21]. LBVS methods can be broadly classified into two categories: similarity searching and pharmacophore modeling[21].

2.3.1 Similarity searching algorithms

Similarity searching algorithms are methods of performing similarity searching on large molecular databases using molecular descriptors or fingerprints and similarity metrics or coefficients. They are used to find molecules that are similar to a reference molecule or a set of reference molecules according to a predefined similarity threshold or cutoff value[21]. There are different types of similarity searching algorithms, such as nearest neighbor search (NNS), clustering, hashing, bit slicing, and others[21].

Nearest neighbor search (NNS) is an algorithm that finds the k most similar molecules to a reference molecule in a database using a similarity metric or coefficient. It can be performed using brute force search, which compares every molecule in the database with the reference molecule, or using indexing structures, such as kd-trees, ball-trees, or cover-trees, which partition the database into smaller subsets based on some criteria[21].

Clustering is an algorithm that groups molecules in a database into clusters based on their similarity using a similarity metric or coefficient. It can be performed using hierarchical clustering, which builds a tree-like structure that shows the nested relationships among clusters, or using partitioning clustering, which divides the database into k non-overlapping clusters based on some criteria[21].

Hashing is an algorithm that maps molecules in a database into hash codes using hash functions that preserve some aspects of similarity between molecules. It can be performed using locality-sensitive hashing (LSH), which uses random projections to map similar molecules into nearby buckets with high probability, or using min-hashing, which uses permutations to map similar molecules into identical buckets with high probability[21].

Bit slicing is an algorithm that slices binary fingerprints into bit slices that correspond to different features or substructures of molecules. It can be performed using inverted indexes, which store lists of molecule identifiers for each bit slice, or using bitmaps, which store bit arrays for each bit slice[21].

The choice of similarity searching algorithms depends on the type and size of the molecular database, the type and format of the molecular descriptors or fingerprints, the type and value of the similarity metric or coefficient, the desired level of efficiency and accuracy of the virtual screening process. Each type of similarity searching algorithm has its own advantages and disadvantages. For example, NNS is simple and intuitive, but it can be slow and memory-intensive for large databases. Clustering is fast and scalable, but it can be sensitive to noise and outliers. Hashing is efficient and compact, but it can introduce false positives and negatives. Bit slicing is flexible and powerful, but it can be complex and redundant[21].

2.3.2 Pharmacophore modeling methods

Pharmacophore modeling methods are methods of generating and using pharmacophore models for virtual screening. Pharmacophore models are abstract representations of the key functional groups or features that are responsible for molecular recognition and interaction. They are used to find molecules that match the pharmacophore features and constraints. There are different types of pharmacophore modeling methods, such as common feature pharmacophore generation, pharmacophore alignment, pharmacophore query generation, pharmacophore screening, and others[21].

Common feature pharmacophore generation is a method of deriving pharmacophore models from a set of known active ligands using feature extraction and selection algorithms. It can be performed using consensus methods, which find common features among all ligands, or using diversity methods, which find diverse features among subsets of ligands[21].

Pharmacophore alignment is a method of aligning pharmacophore models or ligands based on their pharmacophore features using scoring functions that measure their fit and complementarity. It can be performed using rigid alignment methods, which assume fixed conformations for models or ligands, or using flexible alignment methods, which allow

conformational changes for models or ligands[21].

Pharmacophore query generation is a method of creating pharmacophore queries from pharmacophore models or ligands using query optimization algorithms that adjust their features and constraints. It can be performed using heuristic methods, which use empirical rules or criteria to optimize queries, or using statistical methods, which use data-driven approaches to optimize queries[21].

Pharmacophore screening is a method of screening molecular databases for compounds that match pharmacophore queries using matching algorithms that compare their features and constraints. It can be performed using exact matching methods, which require full correspondence between queries and compounds, or using fuzzy matching methods, which allow partial correspondence between queries and compounds[21].

2.4 Chemical descriptors

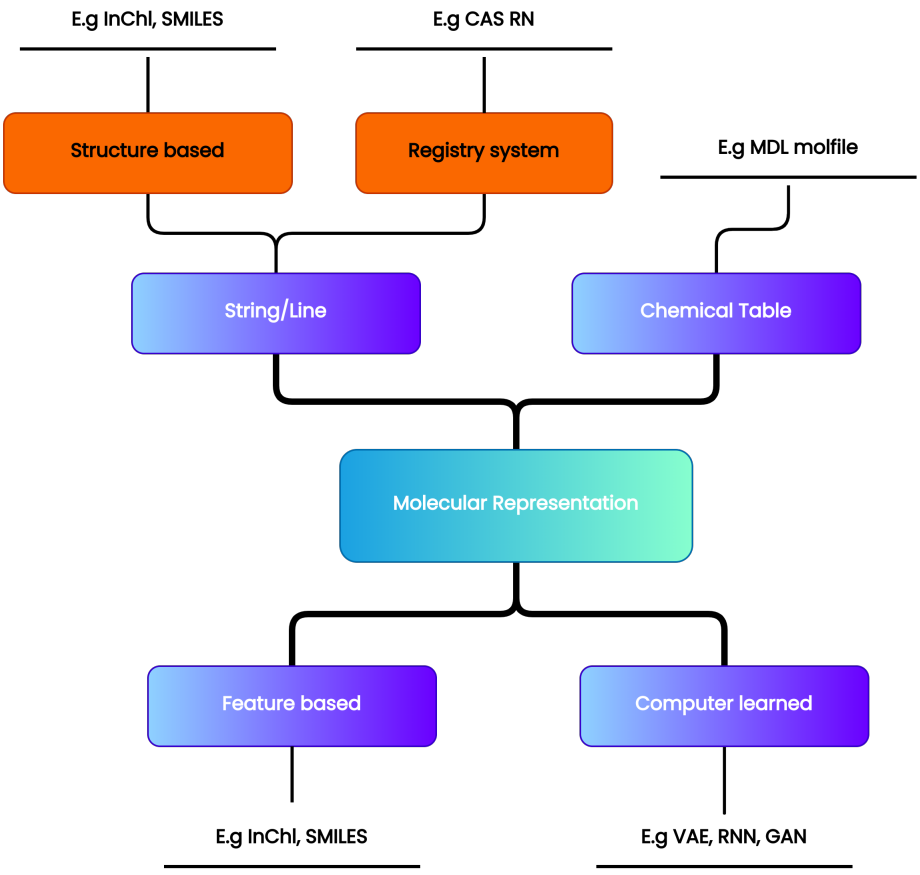


Figure 2.2: Overview of the different classes of molecular representations [12].

Molecular representations are essential for various computational applications in drug discovery, material science, and chemical informatics. A molecular representation is a mathematical or computational encoding of a molecule's structure, properties, or behavior.

2.4.1 String/Line-Based Representations

String or line-based representations are one of the most widely used molecular representations. They are based on the idea of encoding a molecule's structure as a string of characters or a line of text. There are two types of string/line-based representations: structure-based and registry-based.

2.4.1.1 Structure-Based Representations

Structure-based representations are based on the molecular structure of a molecule. They provide a unique representation of the molecule that can be easily exchanged between different computational programs. Some examples of structure-based representations include InChI (International Chemical Identifier) and SMILES (Simplified Molecular Input Line Entry System) [22, 23].

- **InChI** is a standardized and non-proprietary identifier for chemical substances. It represents the molecular structure of a compound as a unique sequence of characters that can be used to identify the compound unambiguously. InChI is widely used in chemical databases, such as PubChem and ChemSpider.
- **SMILES** is a more compact and human-readable representation of a molecule's structure. It encodes the molecular structure as a line of text, where each atom and bond is represented by a unique symbol. SMILES is widely used in chemical informatics, drug discovery, and material science.

2.4.1.2 Registry-Based Representations

Registry-based representations are based on a registry or database of chemical substances. They provide a unique identifier for a molecule that can be used to retrieve its structure and properties from a database. The most widely used registry-based representation is the CAS RN (Chemical Abstracts Service Registry Number) [24].

CAS RN is a unique identifier assigned to every chemical substance registered in the CAS database. It consists of three parts: a prefix, a sequence number, and a check digit. CAS RN is widely used in chemical databases, such as SciFinder and Reaxys.

2.4.2 Chemical Table-Based Representations

Chemical table-based representations are based on the idea of encoding a molecule's structure as a table of data. The most widely used chemical table-based representation is the MDL molfile [25].

MDL molfile is a file format used to store molecular structures, properties, and other information. It consists of several sections, including the connection table, the list of atoms, and the list of bonds. MDL molfile is widely used in computational chemistry, molecular modeling, and drug discovery.

2.5 Feature-Based Representations

Feature-based representations are an essential tool in the field of molecular modeling and drug design. They allow encoding a molecule's properties or behavior as a set of features, which can be used for various applications such as similarity search, clustering, virtual screening, and many others. Among these representations, one of the most widely used is the Morgan fingerprints, also known as circular and topological fingerprints [26].

The Morgan fingerprints are obtained by adapting the standard Morgan algorithm and can be equated to extended-connectivity fingerprints (ECFPs) [27]. Moreover they bring several advantages over other feature-based representations. For instance, they offer fast computation and do not require predefinition, which means they can represent an infinite number of different molecular features. Additionally, they contain chiral information, with each element in the fingerprint representing a specific substructure [26]. The Morgan fingerprints are also easily analyzed and interpreted, and they can be modified accordingly to different needs.

Originally, the Morgan fingerprints were designed to search for molecular features related to activity rather than substructure searches [28]. However, they have since been applied to various areas of molecular modeling and drug design. The fingerprint generation process involves three main steps. Firstly, the atom initialization assigns an integer identifier to each atom. Secondly, the iterative update takes each heavy atom as the center and merges it with the surrounding circle of heavy atoms until the specified radius is reached. Finally, the feature generation performs operations on substructures and generates a list of features.

In contrast to ECFPs, which capture precise substructure details, functional class fingerprints (FCFPs) are more generalized, allowing the same class of functional groups to be used as a feature structure. The Morgan fingerprints are widely used in various areas of molecular modeling and drug design, where they provide a powerful representation of

molecular properties [27]. Overall, the Morgan fingerprints offer a versatile, fast, and effective tool for representing molecular features and are essential for various applications in the field of molecular modeling and drug design.

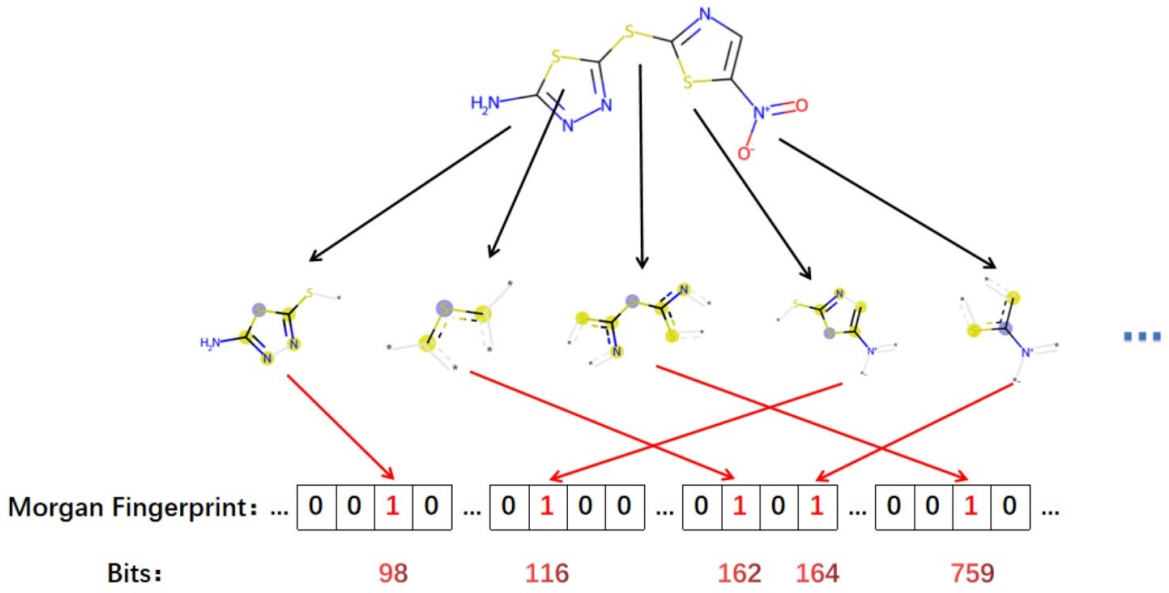


Figure 2.3: Schematic diagram of Morgan fingerprint generation [29].

The Figure 2.3 shows a 2D structure representation of a molecule at the top. The molecule has several atoms connected by bonds and some of them have attached hydrogen atoms. Below the molecule, there are some sub-structures that are extracted from the molecule based on a radius of one bond around each atom. Each sub-structure is assigned a hash value based on its SMILES representation. Below the sub-structures, there is a bit array that represents the Morgan fingerprint of the molecule. The bit array has zeros and ones indicating the presence or absence of each sub-structure in the molecule. The length and order of the bit array depend on the hash values of the sub-structures.

2.5.1 Extended Connectivity Fingerprints (ECFP)

Extended Connectivity Fingerprints (ECFPs) are a popular type of molecular descriptor used in cheminformatics. One of their key features is their ability to represent molecular structures using circular atom neighborhoods [30]. This means that they focus on local structure, rather than the overall structure of the molecule [31]. ECFPs can be very rapidly calculated, which is important when analyzing large datasets.

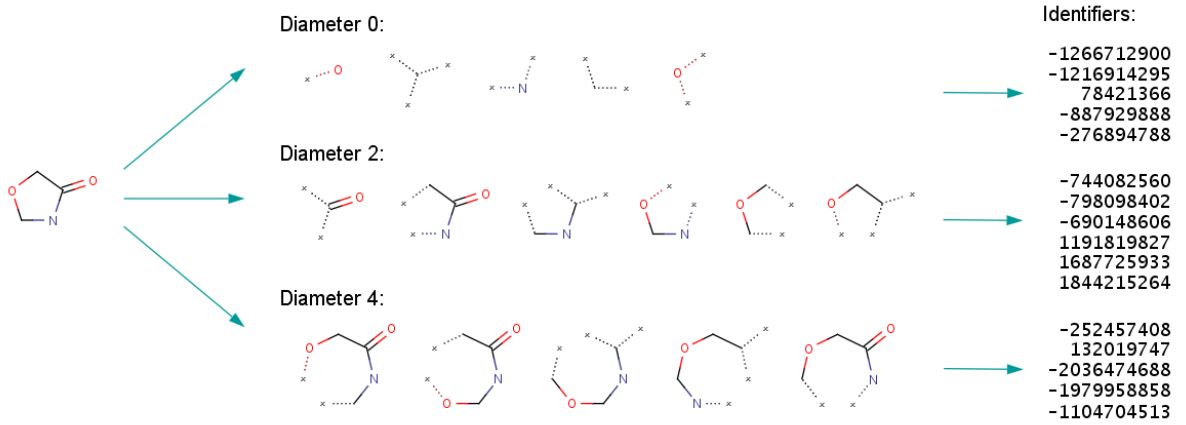


Figure 2.4: ECFP generation process [32].

ECFPs are generated in such a way that their features represent the presence of particular substructures. This is useful for identifying molecular features that are associated with certain biological activities or properties [33]. Another important feature of ECFPs is that they are not predefined and can represent a huge number of different molecular features. This includes stereochemical information, which is important for understanding the three-dimensional structure of molecules.

ECFPs are designed to represent both the presence and the absence of functionality, since both are crucial for analyzing molecular activity. They can be used for a variety of applications, such as virtual screening, lead optimization, and drug discovery. Their generation method can be flexibly customized to produce various types of circular fingerprints for diverse applications [34]. Figures 2.4 and 2.5 illustrates the process of creating a bit-string representation of a chemical structure.

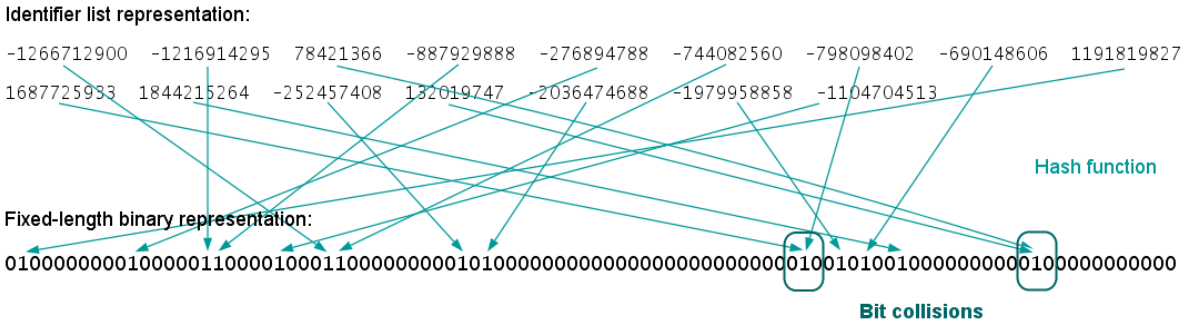


Figure 2.5: Generation of binary vector representation using calculated identifiers [32].

2.6 Molecular Similarity Measurement

2.6.1 Structural Similarity

There are several methods for measuring molecular similarity, but one of the most commonly used approaches is based on the structural features of molecules. This method involves comparing different aspects of two molecules, such as their functional groups, number of bonds, and how atoms are arranged within them. By examining these attributes, scientists can determine how similar or dissimilar two molecules are from one another.

Various algorithms have been created to help measure molecular similarity using this approach. These algorithms take into account a host of factors when comparing two molecules' structures ensuring that the comparison is as precise as possible. This approach has proven to be useful in many different fields including drug discovery and chemical engineering where having an understanding of similarities between various chemicals can be extremely helpful.

The provided table (Table 2.2) presents various commonly used molecular similarity measures along with their corresponding formulas for continuous and binary values. These measures are essential in quantitatively assessing the degree of similarity or difference between two different molecules. The Tanimoto coefficient, also known as the Jaccard index, is computed by dividing the sum of products of features present in both molecules by the sum of squares of features present in either molecule. Similarly, Dice's coefficient or Hodgkin index is calculated using twice the sum of products divided by the sum of squares of features in each molecule alone. On the other hand, Cosine similarity is based on calculating the dot product between two feature vectors and normalizing them with their respective magnitudes. Euclidean distance measures the straight-line distance between two points in space and can be computed using the square root of squared differences between feature values for each molecule. Hamming or Manhattan distance calculates the absolute difference between feature values for each corresponding pair followed by summation across all features to obtain the total distance metric. Soergel's distance metric takes into account the maximum value among pairs when computing the percentage difference in feature values for both molecules from each corresponding pair. Lastly, Russell/Rao similarity measure calculates the ratio between a total number of common features and a total number of possible features while the Forbes similarity measure computes product-moment correlation coefficient scaled by size n to account for differences in overall sizes among molecules being compared.

Coefficient	Continuous Value Formula	Binary Value Formula
Tanimoto (Jaccard)	$Sim_{AB} = \frac{\sum_{i=1}^{N} W_{iA}W_{iB}}{\sum_{i=1}^{N}(W_{iA})^2 + \sum_{i=1}^{N} W_{iB}^2 - \sum_{i=1}^{N} W_{iA}W_{iB}}$	$Sim_{AB} = \frac{c}{a+b-c}$
Dice (Hodgkin)	$Sim_{AB} = \frac{2\sum_{i=1}^{N} W_{iA}W_{iB}}{\sum_{i=1}^{N}(W_{iA})^2 + \sum_{i=1}^{N}(W_{iB})^2}$	$Sim_{AB} = \frac{2c}{a+b}$
Cosine (Carbó index)	$Sim_{AB} = \frac{\sum_{i=1}^{N} W_{iA}W_{iB}}{\sqrt{\sum_{i=1}^{N}(W_{iA})^2}\sqrt{\sum_{i=1}^{N}(W_{iB})^2}}$	$Sim_{AB} = \frac{c}{\sqrt{ab}}$
Euclidean distance	$Dist_{AB} = \sqrt{\sum_{i=1}^{N}(W_{iA} - W_{iB})^2}$	$Dist_{AB} = \sqrt{a+b-2c}$
Hamming (Manhattan) distance	$Dist_{AB} = \sum_{i=1}^{N} \lvert W_{iA} - W_{iB} \rvert$	$Dist_{AB} = a+b-2c$
Soergel distance	$Dist_{AB} = \frac{\sum_{i=1}^{N} \lvert W_{iA} - W_{iB} \rvert}{\sum_{i=1}^{N} \max(W_{iA}, W_{iB})}$	$Dist_{AB} = \frac{a+b-2c}{a+b-c}$
Russell/Rao	$Sim_{AB} = \frac{\sum_{i=1}^{N} W_{iA}W_{iB}}{n}$	$Sim_{AB} = \frac{c}{n}$
Forbes	$Sim_{AB} = \frac{n\sum_{i=1}^{N} W_{iA}W_{iB}}{\sum_{i=1}^{N}(W_{iA})^2 \sum_{i=1}^{N}(W_{iB})^2}$	$Sim_{AB} = \frac{na}{(a+b)(a+b)}$

Table 2.2: Common molecular similarity measures with their formulas [35].

Considering two molecules, A and B, which are characterized by two vectors W_A and W_B of N features. The similarity between molecules A and B is represented by the value S_{AB}. For molecules described by bit-string descriptors, the degree of similarity is calculated based on the number of bits that are set to "1" in the bit-strings of molecules A and B. The number of such bits is represented by a and b respectively. The number of bits that are set to "1" in both bit-strings of molecules A and B is represented by c. The number of bits that are set to "0" in both bit-strings of molecules A and B is represented by d. The total number of bits in both bit-strings of molecules A and B is represented by n, where n = a + b + c + d. For molecules described by descriptors of continuous values, the molecular space is defined by an M x N matrix, where entry W_{ij} is the value of the i_{th} feature ($1 \leq i \leq N$) in the j_{th} molecule ($1 \leq j \leq M$).

Various molecular similarity measures and distance metrics are available in the scientific field to accurately ascertain the resemblance between molecules. However, it is vital to consider multiple factors, including the kind of data used (binary or continuous), dataset size, and research question being investigated while selecting an appropriate measure. It should be noted that no single metric can serve as a perfect measure for all applications. Therefore, employing several methods for analyzing molecular similarities may be necessary to gain a comprehensive understanding of how molecules differ from one another. The use of diverse methods would enable researchers to identify disparities in their studies more effectively and draw more sound conclusions about their findings.

2.7 Machine Learning

Machine learning, a subfield of artificial intelligence, encompasses three main branches: supervised learning, unsupervised learning, and reinforcement learning. Supervised learning focuses on training models with labeled data, unsupervised learning involves discovering patterns in unlabeled data, and reinforcement learning emphasizes on training agents to interact with an environment through rewards and penalties.

2.7.1 Supervised Learning

Supervised learning is a machine learning paradigm where an algorithm learns a function that maps an input to an output based on a given dataset of input-output pairs. The input x and output y form (x, y) pairs that provide supervision for the learning process, and the algorithm aims to find an estimate of the underlying function that generated the data, $\hat{f}$. The supervision signal is calculated as the error between the current estimate and the true label. Supervised learning can be divided into two main types of problems:

regression and classification. Regression problems involve predicting a continuous value, such as the price of a house or the height of a person. Classification problems involve predicting a discrete value, such as the type of animal or the sentiment of a text. The simplest form of regression is linear regression, where the function $\hat{f}$ is assumed to be a linear combination of the input features.

A key component of supervised learning is the loss function, which quantifies the quality of the prediction by comparing the current estimate and the true label. A loss function such as mean squared error can be used to adjust the model parameters to minimize the error and fit the function $\hat{f}$ to the data. In classification problems, the loss function typically measures the probability of assigning the correct label to an input. A common loss function for classification is cross-entropy loss, which penalizes low probabilities for correct labels and encourages high probabilities for incorrect labels.

2.7.2 Unsupervised Learning

In the absence of labeled data, distinct learning algorithms must be employed to facilitate the learning process. The absence of labels necessitates the adoption of unsupervised learning. Unsupervised learning entails the use of an intrinsic metric of the data points, such as distance, for the purpose of learning. Unsupervised learning algorithms can be categorized into three main types: clustering, association, and dimensionality reduction.

- Clustering is a technique that groups data points based on their similarities or differences. Clustering algorithms can be divided into exclusive, overlapping, hierarchical, and probabilistic methods.
- Association is a technique that discovers rules or patterns that describe the relationships between variables or items in a dataset. Association algorithms can be divided into frequent pattern mining, association rule mining, and sequential pattern mining.
- Dimensionality reduction is a technique that reduces the number of features or variables in a dataset while preserving the essential information or structure. Dimensionality reduction algorithms can be divided into linear and nonlinear methods.

2.7.3 Reinforcement learning

Reinforcement Learning is a machine learning domain alongside unsupervised learning (UL) and supervised learning (SL). It is centered on the idea of training an agent to behave intelligently in an environment where the goal is to optimize a progressive reward.

an agent has to deal with the environment through observations or states obtained after performing actions. Therefore his behavior will be improved according to feedback (reward and punishment). The agent keeps updating his way of acting until reaching a goal relating to the state of the environment. These interactions that the agent has to make must be sequential. as the decision made by him must be always dependent on both the previous state and action [36].

Compared to Supervised Learning. RL doesn't need labeled data where the task is to approximate a function that maps a set of inputs into outputs. SL isn't adequate for learning from interaction. Moreover in interactive learning, it is frequently impractical to obtain correct samples of all the situations in which the agent has to act [5].

In addition, RL differs from unsupervised learning on many points. although it often can be thought of as UL and RL both do not require labeled data. The main objective behind RL is to accumulate rewards through a series of actions[5].

2.7.3.1 Formal Framework : Markov Decision Process

Named after the Russian mathematician Andrey Markov, Markov decision processes are the mathematical framework form of reinforcement learning problems. they allow us to make precise theoretical aspects of the RL problem. It's used to model decision-making problems in stochastic, discrete, and sequential environments.

The goal of MDPs is to learn a policy that maximizes the expected cumulative reward over time. Consequently, They are widely used in problems where an agent has to interact with an environment modeled as an MDP and therefore make decisions through those interactions.

Notion of Agent-Environment

In MDPs. The learner is often called the *agent*. Moreover, the entity accommodates everything around the *agent* and interchanges with it. is known as *environment*. These interactions will happen frequently. the agent performing actions while the environment replies to these actions by providing new situations to the agent. Additionally, it gives rewards that the agent wants to maximize over time[5].

The Markov property

A decision process to be considered an MDP must fulfill the Markov property, which means that the next state made by the process must depend only on the current state and the permissible actions in it, and no past memory of previous states can influence

the choice of the next state [37]. The no-memory property is crucial because without it, reasoning about the current state would be infeasible.

Formalism

The formal description in the thesis will follow [5, 36] conventions. In RL, a Markov decision process is defined as a 5-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, R, \gamma)$:

- $\mathcal{S}$ is a finite set of allowed states for the agent to be in an environment. The initial state is represented as s_0.
- $\mathcal{A}$ is a finite set of actions (often noted as action space) that can be taken by an agent.
- $\mathbf{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is the transition function (a.k.a. probability) to the s_{t+1} state given an action a_t and current state s_t.
- $\mathbf{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathcal{R}$ is the reward function after a transition from state s_t to s_{t+1} by performing an action a_t.
- $\gamma \in [0, 1]$ is the discount factor that determines the worth of future rewards.

In an MDP, the system is fully perceivable. Furthermore, an observation at time step t is equivalent to the state of the environment: $\omega_t = s_t$ [36, 37]. The probability of moving from state s_t to s_{t+1} is specified by the transition function $T(s_t, a_t, s_{t+1})$, while the reward is stated in the form of a delimited reward function, $R(s_t, a_t, s_{t+1}) \in \mathcal{R}$.

2.7.3.2 Policies and Value Function

Estimating a term known as the *value function* is a common aspect of Reinforcement Learning algorithms. its role is to actually estimate the worth of the agent when being in a given state (state-action pairs). a worth of a state is defined in terms of expected return. This later depend ton what action that the agent will take. Consequently a certain way of acting, denoted policy will define these value functions [5].

Conventionally, a ***policy*** maps states to the probabilities of selecting each of the available actions [36]. At time t an agent following policy π, $\pi(a|s)$ is the probability that $A_t = a$ if $S_t = s$.

Policies can be either stochastic or deterministic:

- In deterministic instance, the policy is expressed by $\pi(s) : \mathcal{S} \rightarrow \mathcal{A}$.
- In stochastic instance, the policy is expressed by $\pi(s, a) : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ where $\pi(s, a)$ describe the probability of choosing an action a in a state s.

The ultimate goal of an RL problem is to find a policy $\pi(s,a) \in \Pi$ to optimize a a V-value function (or expected return) $V^{\pi}(s) : \mathcal{S} \to \mathbb{R}$ such that :

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} \mid s_t = s, \pi\right], \tag{2.1}$$

Where :

- $r_t = \underset{a\sim\pi(s_t,\cdot)}{\mathbb{E}} R\big(s_t, a, s_{t+1}\big)$,
- $\mathbb{P}\big(s_{t+1}|s_t, a_t\big) = T(s_t, a_t, s_{t+1})$ with $a_t \sim \pi(s_t, \cdot)$,

the optimal expected return will be defined based on the definition of expected return, which is:

$$V^{*}(s) = \max_{\pi\in\Pi} V^{\pi}(s). \tag{2.2}$$

Similarly there are other interesting functions that can be introduced. The Q-value function that define the value of selecting an action a in state s under a policy π is denoted $Q^{\pi}(s,a) : \mathcal{S} \times A \to \mathbb{R}$ and can be summarized as follows:

$$Q^{\pi}(s,a) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} \mid s_t = s, a_t = a, \pi\right]. \tag{2.3}$$

Likewise the optimal Q-value function can also be defined as :

$$Q^{*}(s,a) = \max_{\pi\in\Pi} Q^{\pi}(s,a). \tag{2.4}$$

Having a V-value function can easily lead us to an optimal policy. Even more having a Q-value function makes it more easier to obtain an optimal one.
Another term called advantage function is also widely used in practice. it can be defined as :

$$A^{\pi}(s,a) = Q^{\pi}(s,a) - V^{\pi}(s). \tag{2.5}$$

This function express the worth of the action a, in contrast of the V-value function under a policy π.

2.7.3.3 Deep reinforcement learning

Deep reinforcement learning is a convergence of deep learning and reinforcement learning. Its objective is to acquire the best possible actions that maximize rewards for all potential states of an environment, including but not limited to the bakery, dance hall, and chess board. To achieve this, the agent interacts with complex, high-dimensional environments, engages in action exploration, and leverages feedback to enhance its learning process.

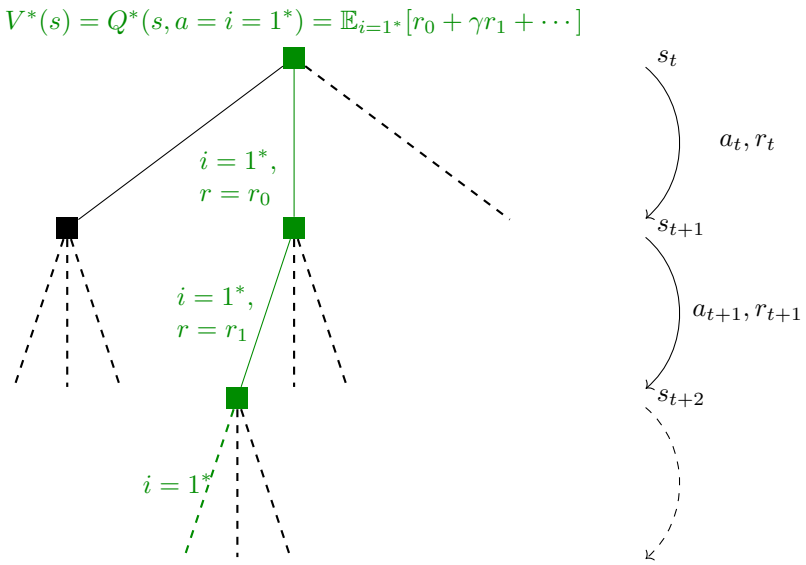


Figure 2.6: Value function illustration.

Value Based methods and policy gradient methods

Value-based methods and policy gradient methods are two prominent approaches in the field of reinforcement learning for acquiring optimal policies. Value-based methods involve the estimation of the value function associated with a particular policy, which is then utilized to enhance the policy by selecting the most favorable action in each state. In contrast, policy gradient methods directly optimize the policy itself, without relying on a value function.

Value-based methods are particularly well-suited for scenarios involving discrete and limited action spaces, where identifying the optimal action is feasible. However, they encounter challenges when confronted with continuous and extensive action spaces, where determining the optimal action becomes computationally expensive or unmanageable [38]. Additionally, value-based methods often learn deterministic policies, which may prove sub-optimal in situations characterized by stochastic environments or requiring exploration.

On the other hand, policy gradient methods prove more effective when addressing problems with continuous and extensive action spaces, where identifying the optimal action is difficult or unattainable. These methods facilitate the learning of stochastic policies, enabling the specification of a probability distribution over actions, which in turn allows for exploration and adaptation within stochastic environments. Policy gradient methods optimize the policy directly by following the gradient of an objective function that quantifies the expected return.

Policy gradient methods exhibit several advantages over value-based methods, including improved convergence properties, reduced variance, and higher sample efficiency.

However, they also possess certain limitations, such as slower learning speed, increased bias, and difficulties in handling discrete actions. Consequently, the selection between value-based methods and policy gradient methods depends on the specific characteristics of the problem at hand and the trade-offs that exist between performance and complexity [38].

Policy gradient algorithms

Policy gradient methods are a class of reinforcement learning algorithms that aim to optimize the parameters of a policy directly through gradient-based optimization. These methods have gained significant attention in recent years due to their effectiveness in addressing high-dimensional and continuous action spaces, offering promising applications in various domains.

Trust region policy optimization (TRPO)

Trust region policy optimization (TRPO) [39] is an algorithm that aims to reduce the high variability in policy parameters. This is achieved by using a special loss function with an additional constraint on the optimization problem. A naive approach to speed up an algorithm is to increase the step size of hyper-parameters such as the learning rate and the policy parameters. However, this approach may fail to uncover solutions hidden in finer grained trajectories and result in convergence to local optima. TRPO avoids this by using adaptive step size that depends on the output of the optimization progress. Trust regions are used to constrain the update size in general optimization problems. The algorithm computes the quality of the approximation and expands the trust region if it is still good. Alternatively, the trust region can be shrunk if the divergence of the new and current policy is getting large [37]. TRPO maximizes the loss function subject to the constraint that the old and the new policies are not too far away from each other. It uses the Kullback–Leibler divergence for this purpose. The algorithm scales to complex high-dimensional problems and has demonstrated its robust performance in simulated robotic Swimming, Hopping, Walking gaits, and Atari games. However, TRPO is a complicated algorithm that uses second-order derivatives, and its implementation can be challenging.

Proximal Policy Optimization (PPO)

Proximal Policy Optimization (PPO) is a learning algorithm that was designed as an enhancement of TRPO [39]. PPO shares several advantages of TRPO, but is less complex to implement, more versatile, exhibits better empirical sample complexity, and has a

superior run-time complexity. The fundamental aim of PPO, akin to TRPO, is to determine the largest possible improvement in the policy parameter, while avoiding a collapse in performance [37].

PPO has two distinct versions: PPO-Penalty and PPO-Clip. The former resolves a KL-constrained update akin to TRPO, but instead of a hard constraint, it imposes a penalty on the KL-divergence in the objective function. On the other hand, PPO-Clip doesn't use a KL-divergence term in the objective function and doesn't have any constraints. It depends on clipping within a fixed range $[1-\epsilon, 1+\epsilon]$. At in the objective function to eliminate incentives for the new policy to deviate significantly from the old policy.

The authors of PPO [6] offer an implementation of PPO as a reference point (See Algorithm 1). Both TRPO and PPO are on-policy algorithms, and their design choices are reflected upon by Hsu et al (2020) [40].

Algorithm 1 PPO-Clip

Input: initial policy parameters θ_0, initial value function parameters ϕ_0
for $k = 0, 1, 2, \ldots$ **do**
  Collect set of trajectories $\mathcal{D}_k = \{\tau_i\}$ by running policy $\pi_k = \pi(\theta_k)$ in the environment.
  Compute rewards-to-go $\hat{R}_t$.
  Compute advantage estimates, $\hat{A}_t$ (using any method of advantage estimation) based on the current value function V_{ϕ_k}.
  Update the policy by maximizing the PPO-Clip objective:

$$\theta_{k+1} = \arg\max_{\theta} \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \min\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)} A^{\pi_{\theta_k}}(s_t, a_t),\ \ g(\epsilon, A^{\pi_{\theta_k}}(s_t, a_t)) \right),$$

  typically via stochastic gradient ascent with Adam.
  Fit value function by regression on mean-squared error:

$$\phi_{k+1} = \arg\min_{\phi} \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \left(V_\phi(s_t) - \hat{R}_t \right)^2,$$

  typically via some gradient descent algorithm.
end for

Actor-Critic in PPO

In the realm of deep reinforcement learning, the actor-critic method plays a vital role in enhancing the performance of agents. This approach combines the strengths of both value-based and policy-based methods by employing two distinct components: the actor and the critic. The actor is responsible for selecting actions based on the current policy, while the critic evaluates the value of state-action pairs. In the context of Proximal Policy

Optimization (PPO), the actor-critic framework facilitates the optimization process by leveraging the advantages of both exploration and exploitation. The actor explores the environment by sampling actions and estimating their quality, while the critic provides feedback on the value of these actions, enabling the actor to adjust its policy accordingly. By incorporating both the actor and the critic, the PPO algorithm benefits from a more robust and efficient learning process, resulting in improved decision-making and policy optimization in complex environments.

2.8 Convolutional Neural Network (CNN) architecture

Convolutional neural networks (CNNs) are a category of artificial neural networks frequently employed for visual imagery analysis [41] within the domain of deep learning. CNNs substitute the general matrix multiplication with the convolution operation in at least one of their layers [42], and are intended to process pixel data. As such, they find extensive application in image recognition and processing , image and video recognition, recommender systems [43], image classification, image segmentation, medical image analysis, natural language processing [44], brain-computer interfaces [45], and financial time series [46].

The convolutional architecture of CNNs is based on the shared-weight structure of the convolution kernels or filters, which slide along input features and provide translation-equivariant responses known as feature maps [47, 48]. Despite being referred to as Shift Invariant or Space Invariant Artificial Neural Networks (SIANN), most CNNs are not actually invariant to translation due to the downsampling operation applied to the input [49]. Figure 2.7 depict an example of how CNN architecture works in image classification.

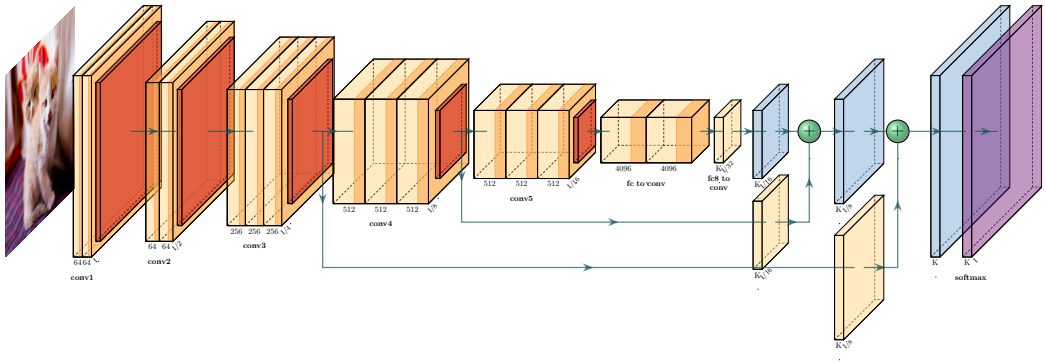


Figure 2.7: A Convolutional Neural Network (CNN) architecture for image classification [50].

CNNs are a type of regularized multi-layer perceptron that addresses the issue of overfitting. Multilayer perceptrons are fully connected networks that can suffer from overfitting, but CNNs take a different approach to regularization. They leverage the hierarchical pattern in data to assemble patterns of increasing complexity using smaller and simpler patterns in their filters. This hierarchical approach enables CNNs to efficiently learn complex patterns in data while minimizing the risk of overfitting.

Convolutional networks were inspired by biological processes [51–54], specifically the organization of the animal visual cortex. The receptive fields of individual cortical neurons partially overlap, such that they cover the entire visual field. CNNs use relatively little pre-processing compared to other image classification algorithms and can optimize filters through automated learning, without the need for hand-engineering. This independence from prior knowledge and human intervention in feature extraction is a major advantage.

2.9 Multi-Head attention architecture

The multi-head attention architecture [55] is a powerful technique utilized in deep learning models that allows for efficient processing of input data, including but not limited to sentences and paragraphs. This technique employs a unique approach by dividing the input data into multiple "heads," or segments (see Figure 2.8), which can each attend to different parts of the input simultaneously. By doing so, this architecture enables the model to effectively capture dependencies between distant parts of the input sequence, leading to a more comprehensive analysis. The attention head operation can be summarized in this equation.

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.6}$$

The concept of scaled dot product attention enables a neural network to selectively focus on different aspects of a sequence. However, in many cases, a sequence element may need to attend to multiple distinct aspects, making a single weighted average insufficient. Consequently, it becomes necessary to expand the attention mechanism to incorporate multiple heads, each involving separate query-key-value triplets operating on the same set of features. Specifically, by transforming the query, key, and value matrices into h sub-queries, sub-keys, and sub-values, we can independently apply the scaled dot product attention mechanism to each of them. Subsequently, the resulting heads are concatenated and combined using a final weight matrix. This operation can be expressed mathematically as follows:

$$\text{Multihead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_h)W^O \tag{2.7}$$

where

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{2.8}$$

The parameters $W_{1...h}^Q \in \mathbb{R}^{D \times d_k}, W_{1...h}^K \in \mathbb{R}^{D \times d_k}, W_{1...h}^V \in \mathbb{R}^{D \times d_v}$, and $W^O \in \mathbb{R}^{h \cdot d_v \times d_{out}}$ are learned. Here, D represents the input dimensionality, while d_k and d_v represent the dimensions of the key and value vectors, respectively. The final weight matrix, W^O, has a dimension of $h \cdot d_v \times d_{out}$.

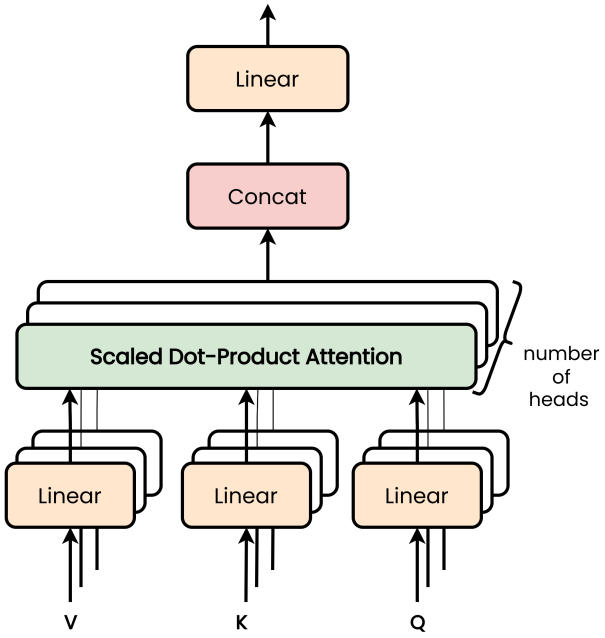


Figure 2.8: Overview of multi-head attention architecture.

Moreover, when combined with other architectural features like convolutional layers, multi-head attention can result in highly effective and efficient deep learning models. This combination maximizes the benefits of both techniques and produces accurate results quickly.

The true advantages of multi-head attention become apparent when applied to natural language processing tasks where long sequences and complex structures are common. These tasks require intricate analyses that involve multiple levels of abstraction which traditional methods often struggle with. However, due to its ability to attend to different parts of the input simultaneously and capture dependencies between them effectively, multi-head attention-based deep learning models excel at these tasks.

In conclusion, multi-head attention architecture is an innovative way for deep learning models to process input data efficiently and accurately while capturing dependencies between distant parts of the sequence effectively; this makes it ideal for natural language

processing tasks. Its benefits are amplified when combined with other architectural features such as convolutional layers resulting in highly effective and efficient deep learning models that produce accurate results quickly.

2.10 Related work

In this section we will present and discuss some state-of-the-art work that involves using fingerprint and machine learning methods to enhance ligand-based virtual screening performance in widely used dataset.

2.10.1 Bayesian inference network (BIN) in LBVS

Bayesian inference network (BIN) is a probabilistic graphical model that represents the dependencies and independencies among a set of variables using nodes and edges. BIN can be used for ranking a database in order of decreasing probability of bioactivity based on a query molecule and its features. BIN has been applied to molecular similarity searching, which is a ligand-based virtual screening technique that aims to find molecules that are similar to a given query molecule based on their structural features.

One of the early studies that used BIN for molecular similarity searching is by Chen et al [56], who introduced a novel similarity searching approach using a BIN, where each node represents a molecular feature and each edge represents the conditional probability between two features. They used four different descriptors (ECFP-4, MACCS, PubChem and Unity) for the MDDR dataset (DS1, DS2 or DS-HOM, and DS3 or DS-HET), and computed the probability of bioactivity for each database molecule using the Bayesian inference formula. They compared their method with the conventional Tanimoto similarity method (TAN), and showed that their method provided an interesting alternative to TAN, especially when the actives molecules being sought have a high degree of structural homogeneity.

Abdo et al [57] proposed a method to implement relevance feedback in ligand-based virtual screening using Bayesian inference network (BIN). They used the feedback information from the top-ranked structures of unknown activity to form a set of active reference structures, which were then used to enhance the BIN search in two ways: reweighting the fragments in the reference structures and group fusion techniques. They evaluated their method on three MDL Drug Data Report datasets and showed that it improved the cost-effectiveness of BIN, especially for higher diversity datasets. They also compared their method with other relevance feedback methods based on Tanimoto similarity and found that BIN performed better in most cases.

The article "Ligand-Based Virtual Screening Using Bayesian Inference Network" by Chen et al [58] presents a novel approach for finding novel bioactive molecules by searching large chemical databases for structures that are similar to a known active compound. The authors propose to use a Bayesian inference network (BIN), a probabilistic graphical model that can represent the conditional dependencies between molecular features and bioactivity, and can rank the database molecules in order of decreasing probability of activity. The authors compare their approach with a conventional similarity method based on the Tanimoto coefficient, using two popular chemoinformatics databases: the MDL Drug Data Report database (MDDR) and the World of Molecular Bioactivity database (WOMBAT). The authors show that their approach outperforms the conventional method, especially when the active molecules have a high degree of structural homogeneity. However, they also acknowledge that their approach performs poorly when the active molecules are structurally diverse or when the reference structure is not representative of the activity class. The authors suggest some possible ways to improve their approach, such as using relevance feedback or data fusion techniques.

2.10.1.1 Advantages

The BIN has some advantages over other tools for similarity-based virtual screening, such as fingerprint-based methods or machine learning models. The BIN can capture the structural diversity and complexity of molecules better than fixed-length fingerprints, and it can handle missing or noisy data better than machine learning models that require training and validation1. The BIN can also incorporate relevance feedback information from previous screening results to improve its performance.

2.10.1.2 Limitations

However, the BIN also has some limitations and challenges. One of the main challenges is how to select the optimal set of reference molecules for a given screening task. The BIN works best when the reference molecules are structurally homogeneous, meaning that they share a common scaffold or core structure that is responsible for their biological activity. However, when the reference molecules are structurally heterogeneous, meaning that they have different scaffolds or core structures that confer activity, the BIN may fail to capture the essential features that distinguish them from inactive molecules. Another challenge is how to balance the trade-off between specificity and sensitivity in similarity-based virtual screening. The BIN tends to favor specificity over sensitivity, meaning that it can identify molecules that are very similar to the reference molecules but may miss some molecules that are less similar but still active. This may limit the discovery of novel scaffolds or

core structures that have activity.

Therefore, the BIN is a promising tool for similarity-based virtual screening, but it still requires further research and development to overcome its limitations and challenges.

2.10.2 Siamese Networks for LBVS

Siamese networks are a type of neural network architecture that can learn similarity functions between pairs of inputs. They have been applied to various domains such as face recognition, signature verification, and natural language processing. Recently, some studies have explored the use of Siamese networks for ligand-based virtual screening (LBVS), which is a computational technique that aims to identify potential drug candidates based on their similarity to known active molecules.

One of the studies that used Siamese networks for LBVS is by Altalib and Salim [59]. They proposed an enhanced Siamese multi-layer perceptron model that can handle structurally heterogeneous molecules better than the standard Tanimoto coefficient. They reported improved recall and precision scores on several benchmark datasets, including MDDR (DS1, DS2, DS3).

Another study that used Siamese networks for LBVS is by Altalib et al [60]. They proposed a hybrid-enhanced Siamese model that combines different types of deep learning methods, such as convolutional neural networks, recurrent neural networks, and attention mechanisms. They also reported better performance than the standard Tanimoto coefficient and other methods on MDDR (DS1, DS2, DS3) and MUV datasets.

2.10.2.1 Advantages

One of the advantages of using siamese networks for ligand-based virtual screening is their ability to learn complex and nonlinear similarity functions from data. Unlike traditional similarity measures that rely on predefined rules or formulas, siamese networks can learn the optimal similarity function from a large number of pairs of similar and dissimilar molecules. Siamese networks can also adapt to different types of data and tasks by changing their architecture and parameters.

Another advantage of using siamese networks for ligand-based virtual screening is their ability to capture the structural and biological similarity of molecules based on their features and interactions. Siamese networks can extract meaningful features from the molecular representations, such as fingerprints, graphs, or images, and compare them based on their overlap or distance. Siamese networks can also model the interactions between molecules and their targets or environments, such as docking poses, binding affinities, or pharmacophores, and use them to enhance the similarity scores.

A third advantage of using siamese networks for ligand-based virtual screening is their ability to be integrated with other similarity measures or machine learning models to enhance the performance of LBVS. Siamese networks can be combined with other similarity measures, such as fingerprint-based or shape-based methods, to create hybrid or ensemble models that leverage the strengths of each method. Siamese networks can also be integrated with other machine learning models, such as neural networks, support vector machines, or random forests, to create multi-task or multi-modal models that exploit the complementary information from different sources.

2.10.2.2 Limitations

One of the main challenges of using siamese networks for ligand-based virtual screening is the high demand of data and computational resources. Siamese networks need to learn the features of molecules from a large number of pairs of similar and dissimilar molecules, which requires a lot of training data and time. Moreover, siamese networks need to be carefully tuned and optimized to avoid poor performance. The choice of similarity function, loss function, and regularization techniques can have a significant impact on the quality of the learned features and the similarity scores. If the siamese network is too complex or too simple, it may overfit or underfit the data, respectively, leading to inaccurate results.

Another challenge of using siamese networks for ligand-based virtual screening is the difficulty of capturing the structural diversity and complexity of molecules. Siamese networks rely on the assumption that molecules with similar features have similar biological activity. However, this assumption may not hold for all cases, especially when the active molecules have different scaffolds or core structures that are responsible for their activity. Siamese networks may fail to recognize the essential features that distinguish active molecules from inactive ones, or they may assign high similarity scores to molecules that share irrelevant features.

A third challenge of using siamese networks for ligand-based virtual screening is the trade-off between specificity and sensitivity. Siamese networks tend to favor specificity over sensitivity, meaning that they can identify molecules that are very similar to the reference molecules but may miss some molecules that are less similar but still active. This may limit the discovery of novel scaffolds or core structures that have activity. On the other hand, if siamese networks are too sensitive, they may retrieve too many false positives, meaning that they may assign high similarity scores to molecules that are not active.

2.10.3 Convolutional Neural Networks for LBVS

Convolutional neural networks (CNNs) are a type of deep learning model that can extract features from images or graphs by applying convolutional filters. They have been widely used for image recognition, natural language processing, and graph analysis. Recently, some studies have explored the use of CNNs for ligand-based virtual screening (LBVS), which is a computational technique that aims to identify potential drug candidates based on their similarity to known active molecules.

One of the studies that used CNNs for LBVS is by Madjid et al [61]. They proposed a novel CNN architecture that can learn molecular fingerprints from 2D molecular graphs. They compared their model with several baseline methods, such as Tanimoto coefficient, random forest, and graph neural network, on three benchmark datasets: MDDR (DS1, DS2, DS3), MUV, and ChEMBL. They reported that their model achieved the best performance in terms of accuracy, recall, precision, and F1-score.

Some of the benefits of using CNNs for LBVS are:

They can learn complex and nonlinear features from molecular graphs without relying on predefined fingerprints. They can capture the structural and biological similarity of molecules based on their local and global patterns. They can be integrated with other deep learning models or similarity measures to enhance the performance of LBVS. Some of the limitations of using CNNs for LBVS are:

- They require large and balanced datasets to train effectively.
- They may suffer from overfitting or underfitting depending on the network architecture and hyperparameters.
- They may be affected by the choice of molecular representations or reference structures.

2.10.4 Stacked deep belief network (SDBN) in LBVS

Stacked deep belief network (SDBN) is a deep learning method that consists of multiple layers of restricted Boltzmann machines (RBMs), which are stochastic neural networks that can learn the probability distribution of the input data. SDBN can be used for feature extraction, dimensionality reduction, and classification tasks. SDBN has been applied to molecular similarity searching, which is a ligand-based virtual screening technique that aims to find molecules that are similar to a given query molecule based on their structural features.

One of the recent studies that used SDBN for molecular similarity searching is by Nasser et al [62], who proposed a multi-descriptor-based method that reweighted the

molecular features using deep belief networks. They used three different descriptors (ECFC-4, ECFP-4, and EPFP-4) for the MDDR dataset (DS1, DS2, and DS3), each of which represents different important features. They trained SDBN with each descriptor individually to select the important features based on a new weight, with a lower error rate, and merged together all new features from all descriptors to produce a new descriptor for similarity searching. They compared their method with several existing benchmark similarity methods, such as Bayesian inference networks (BIN), the Tanimoto similarity method (TAN), adapted similarity measure of text processing (ASMTP) and the quantum-based similarity method (SQB). They found that their method outperformed the other methods in terms of accuracy, precision, recall, F1-score and ROC-AUC on both internal and external validation sets.

Another study that used SDBN for molecular similarity searching is by Hamza et al [63], who proposed a feature reduction method based on auto-encoder deep learning. They used four different descriptors (ECFC-4, ECFP-4, MACCS and PubChem) for the MDDR dataset (DS1, DS2, and DS3), and applied auto-encoder to remove irrelevant and redundant features from each descriptor. They then used SDBN to reweight the remaining features and generate a new descriptor for similarity searching. They compared their method with TAN, ASMTP and SQB methods, and showed that their method achieved higher accuracy and recall than the other methods on structurally heterogeneous datasets.

A limitation of both studies is that they did not consider the biological activity of the molecules in their similarity searching methods. They only focused on the structural similarity based on 2D fingerprints, which may not capture the 3D conformation and pharmacophore features that are relevant for hit-target identification.

2.11 Conclusion

This chapter has provided an overview of the main concepts and methods in chemoinformatics, molecular representation, similarity measures, and virtual screening. It has also introduced the theoretical foundations and applications of deep reinforcement learning (DRL) in drug discovery. The chapter has reviewed some of the state-of-the-art models that use DRL for ligand-based virtual screening (LBVS), highlighting their strengths and limitations. The chapter has identified some of the research gaps and challenges that motivate the development of novel DRL approaches for LBVS, such as:

- How to design effective and efficient molecular representations that capture both structural and functional information of molecules.
- How to incorporate prior knowledge and domain expertise into DRL models to guide

the exploration and exploitation of the chemical space.

- How to evaluate and compare the performance and robustness of different DRL models for LBVS using appropriate metrics and benchmarks.
- How to improve the interpretability and explainability of DRL models for LBVS to provide insights into the molecular mechanisms and interactions.

The next chapter will present the proposed DRL framework for LBVS, which aims to address some of these challenges and improve the quality and diversity of the generated molecules.

Chapter 3

Research Methodology

3.1 Introduction

As noted in the previous chapter Machine Learning techniques have always caught the interest of researchers in chemoinformatics. Particularly in QSAR/QSPR machine learning techniques have shown great results in various tasks. The demonstrated success of these techniques in various applications has underscored their significance. Furthermore, the continued popularity of ML is evidenced by the ongoing testing and implementation of newly developed techniques [64].

The primary aim of this study was to develop a novel molecular representation by utilizing extracted features from agent policy, with the goal of enabling discrimination between target/query molecules with disparate biological activities.

This chapter provides an overview of the research methodology utilized to achieve the study's objectives and ensure the attainment of its goals.

3.2 Research Design

The research design section presents the framework for conducting the study, consisting of two subsections: Research Framework, which outlines the four phases of the research process, and Datasets, which describes the dataset employed in the study.

3.2.1 Research Framework

As illustrated in Figure 3.1 Our work in this thesis can be split into four main phases as follows :

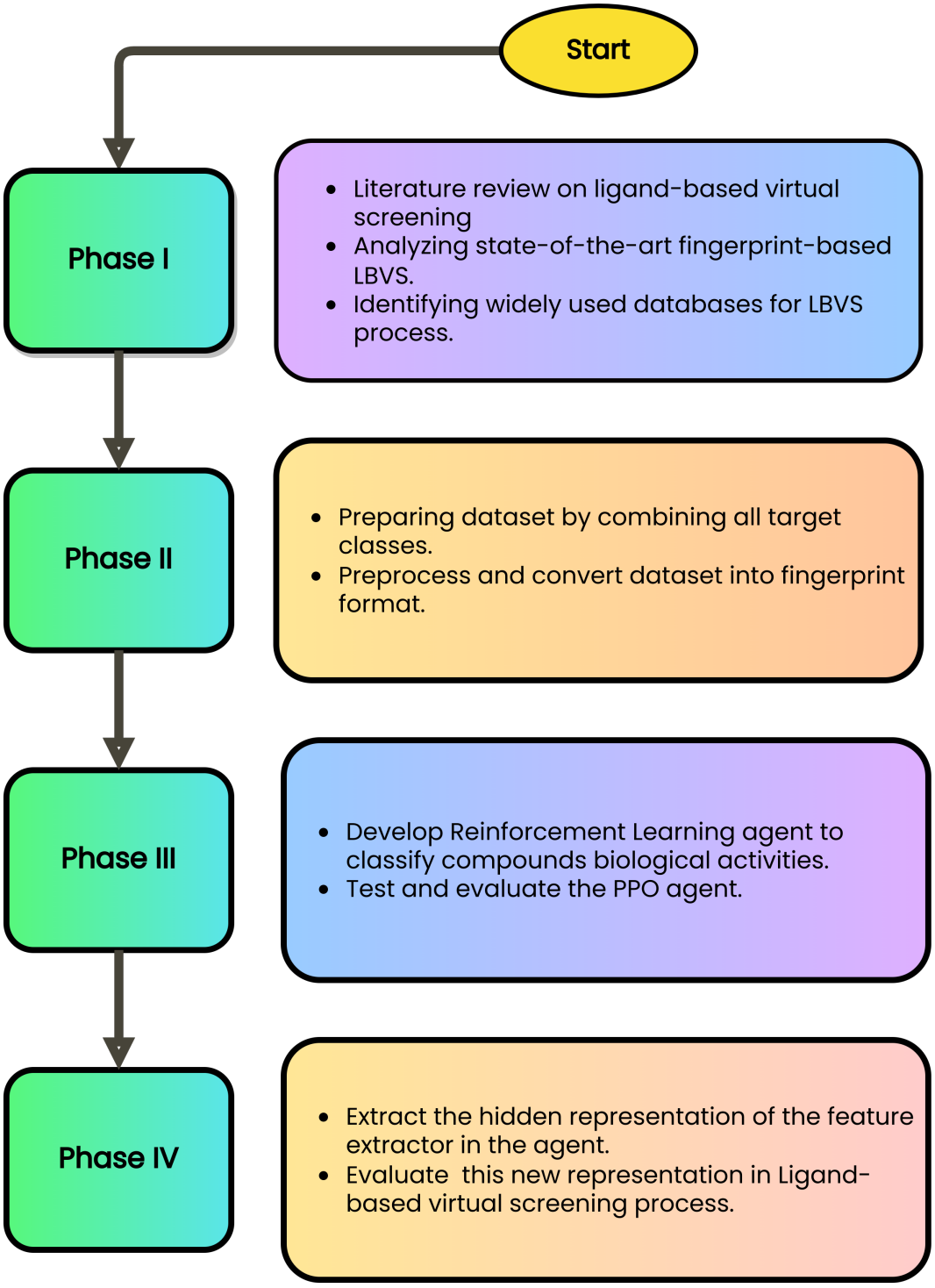


Figure 3.1: Flow Chart illustrating Research Framework's phases

3.2.1.1 Phase I

Reviewing the depiction of molecular representation and similarity measures of state-of-the-art works to point out necessary measures to improve the LBVS process using Reinforcement Learning.

3.2.1.2 Phase II

A benchmark dataset have been composed by fusing all compounds of biological activities into one data set where each molecule have one or more class. This dataset is is then split into two slice, a train part used to train the agent, hence a test part to evaluate agent's performance.

3.2.1.3 Phase III

A comprehensive reinforcement learning framework has been devised. The framework comprises two primary components: a custom environment featuring either the training or test segment of the dataset, and a PPO agent network [6]. The observation space of the environment is a binary vector of 1024 bits, denoting the ECFP4 fingerprint. In contrast, the action space is a binary vector equivalent to the number of biological activities in the dataset.

The PPO agent network has two major components: a feature extractor and an MLP extractor. The feature extractor is comprised of multi-1D-convolution layers, a multi-head attention layer, and a linear layer. Its primary objective is to extract the most important features in the observation. The MLP extractor, on the other hand, takes the last linear layer of the feature extractor and passes it through multiple linear layers. It has two output values: an estimate of the current policy's value function and the predicted action.

Finally, the agent's performance is evaluated using several metrics, including accuracy, by using the test section of the dataset.

3.2.1.4 Phase IV

A ligand-based virtual screening process [65] will be conducted on the dataset utilizing recently extracted feature vectors. Ten queries will be chosen from each biological activity class and screened against actives of the same class along with decoys. The recall score of these selected queries will be recorded to compare with existing state-of-the-art methods.

3.2.2 Datasets

In this section, we provide an overview of the dataset used in our master thesis. The primary dataset utilized for our research is the MDDR (Molecular Data for Drug Discovery and Research) database. MDDR is a comprehensive cheminformatics database widely employed in ligand-based virtual screening (LBVS) studies.

3.2.2.1 MDDR

The MDDR dataset [66] has been a significant tool in the field of virtual screenings. Through experimental simulations, it has been demonstrated that the proposed technique can significantly improve the efficiency of ligand-based virtual screenings. This is particularly beneficial for more diverse datasets, as it provides various means of enhancing their accuracy. The MDDR benchmark datasets (MDDR-DS1, MDDR-DS2, and MDDR-DS3) are three unique types of datasets carefully selected from the comprehensive MDDR database. Each benchmark dataset serves a specific purpose and is tailored to suit different research requirements. Researchers can utilize these datasets to gain insights into various drug discovery processes and effectively evaluate new techniques without compromising on accuracy or quality. Overall, this dataset has proven to be an invaluable resource for researchers looking to advance their knowledge and understanding of drug development processes through virtual screenings.

Within the MDDR (MDL Drug Dataset Report) dataset, researchers have identified a variety of activity classes. In fact, there are eleven different classes included in the MDDR-DS1 alone. Some of these activities share similar structures and therefore can be classified as homogenous, while others possess diverse structures and are deemed heterogeneous. The MDDR-DS2 dataset features ten homogeneous activity classes, which are activities that share structural similarities in their chemical makeup. On the other hand, the MDDR-DS3 data set includes ten heterogeneous activity classes; these activities have varying chemical compositions and thus exhibit differing effects within the body. By categorizing these various activities based on their structure and composition, scientists can better understand how different molecules interact with biological systems to produce specific effects.

- **MDDR-DS1** : composed of 102516 compounds of both active and decoy group where the majority are decoys. There exists 11 molecular activity class where homogeneous and heterogeneous active elements are merged together. Table 3.1 shows activity classes, pairwise similarity of DS1's molecules.

Activity Index	Active Molecules	Activity Class	Pairwise Similarity
31420	1130	Renin inhibitors	0.290
71523	750	HIV protease inhibitors	0.198
37110	803	Thrombin inhibitors	0.180
31432	943	Angiotensin II AT1 antagonists	0.229
42731	1246	Substance P antagonists	0.149
06233	752	5HT3 antagonist	0.140
06245	359	5HT reuptake inhibitors	0.122
07701	395	D2 antagonists	0.138
06235	827	5HT1A agonists	0.133
78374	453	Protein kinase C inhibitors	0.120
78331	636	Cyclooxygenase inhibitors	0.108

Table 3.1: Activity classes, pair-wise similarity, indexes and number of molecules for DS1.

- **MDDR-DS2** : composed of 102516 compounds of both active and decoy group where the majority are decoys. There exists 10 molecular activity class where only homogeneous active elements have been selected. Table 3.2 shows activity classes, pairwise similarity of DS2's molecules.

Activity Index	Active Molecules	Activity Class	Pairwise Similarity
07707	207	Adenosine (A1) agonists	0.229
07708	156	Adenosine (A2) agonists	0.305
31420	1130	Renin inhibitors	0.290
42710	111	CCK agonists	0.361
64100	1346	Monocyclic lactams β	0.336
64200	113	Cephalosporins	0.322
64220	1051	Carbacephems	0.269
64500	126	Carbapenems	0.260
64350	388	Tribactams	0.305
75755	455	Vitamin D analogous	0.386

Table 3.2: Activity classes, pair-wise similarity, indexes and number of molecules for DS2.

- **MDDR-DS3** : composed of 102516 compounds of both active and decoy group where the majority are decoys. There exists 10 molecular activity class where only

heterogeneous active elements have been selected. Table 3.3 shows activity classes, pairwise similarity of DS3's molecules.

Activity Index	Active Molecules	Activity Class	Pairwise Similarity
09249	900	Muscarinic (M1) agonists	0.111
12455	1400	NMDA receptor antagonists	0.098
12464	505	Nitric oxide synthase inhibitors	0.102
31281	106	Dopamine -hydroxylase inhibitors	0.125
43210	957	Aldose reductase inhibitors	0.119
71522	700	Reverse transcriptase inhibitors	0.103
75721	636	Aromatase inhibitors	0.110
78331	636	Cyclooxygenase inhibitors	0.108
78348	617	Phospholipase A2 inhibitors	0.123
78351	2111	Lipoxygenase inhibitors	0.113

Table 3.3: Activity classes, pair-wise similarity, indexes and number of molecules for DS3.

3.3 Design and Steps

In this section, we describe the design and steps of our proposed PPO agent for multi-label classification. Our PPO agent consists of two modules: an actor and a critic. The actor is a stochastic policy network that outputs the conditional probability of taking each action given an observation. The critic is a value function network that estimates the expected discounted cumulative long-term reward for following the policy from the current observation. Both the actor and the critic are implemented as neural networks with three components: a feature extractor, an MLP extractor, and an action/value network.

The feature extractor is composed of a convolutional neural network (CNN) and multi-head attention network that takes an ECFP4 fingerprint as input and extracts high-level features from it. The MLP extractor is a multi-layer perceptron (MLP) that takes the output of the feature extractor and further processes it into a latent vector. The action/value network is a linear layer that takes the output of the MLP extractor and produces either an action distribution (for the actor) or a value estimate (for the critic).

3.3.1 Data preprocessing

The original form of MDDR dataset molecules is smiles representation. although it's one of the most used form of representation in traditional methods we've chosen to use a binary

representation fingerprint to benefit from the architectural choice of using convolutional feature extractor that have been successfully applied on data with similar shapes such as images, text embedding. Figure 3.2 illustrate the process of converting molecule to fingerprint format.

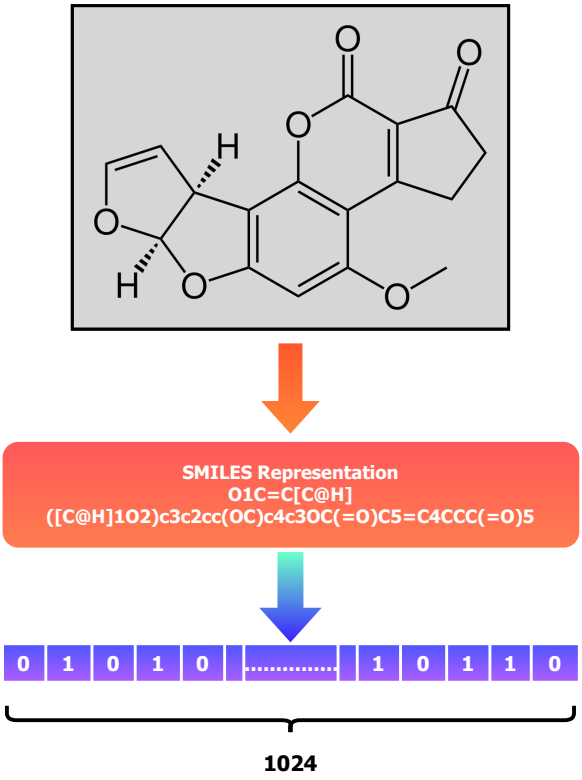


Figure 3.2: Steps of converting molecule data to bit-string fingerprint.

The data is converted using PipeLine Software [67] into text file format where ECFP-4 binary vectors stored in form of string format separated by space. each line corresponds to fingerprint of a molecule.

3.3.2 Merging Dataset

In our work we've only selected active compounds of the MDDR dataset. those active compounds of the three subsets will be merged together into one single database, hence Converting DS1, DS2, DS3 into a single dataset. Each DS partition is loaded using the file `ActivityClass.txt`, this later contains the indexes of all biological activities (11,10,10) Figure 3.4 shows an example how it could be. Thereafter all fingerprint print are loaded and associated into their labels using those indexes.

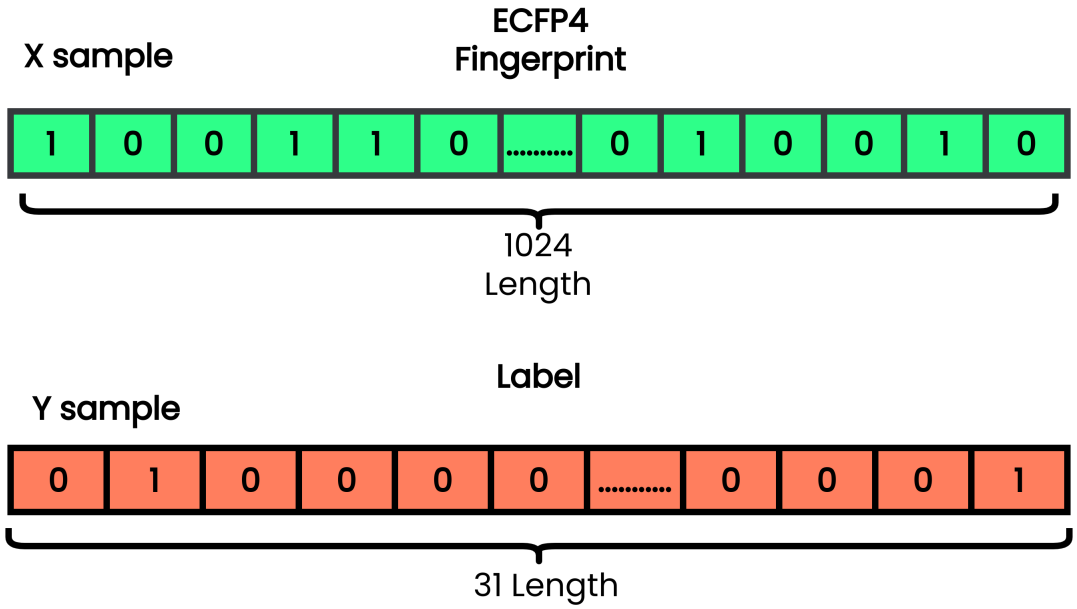


Figure 3.3: A sample of created dataset.

MDDR > DS1 > ActivityClass.txt

```
1    11
2    1   1130   1130
3    1131   1880   750
4    1881   2683   803
5    2684   3626   943
6    3627   4872   1246
7    4873   5624   752
8    5625   5983   359
9    5984   6378   395
10   6379   7205   827
11   7206   7658   453
12   7659   8294   636
```

Figure 3.4: DS1 activity class depicting indexes of biological activities

Moreover after each DS partition the fingerprints and their labels(converted into one hot-encoding) will be concatenated with previous partitions. Finally fingerprints that

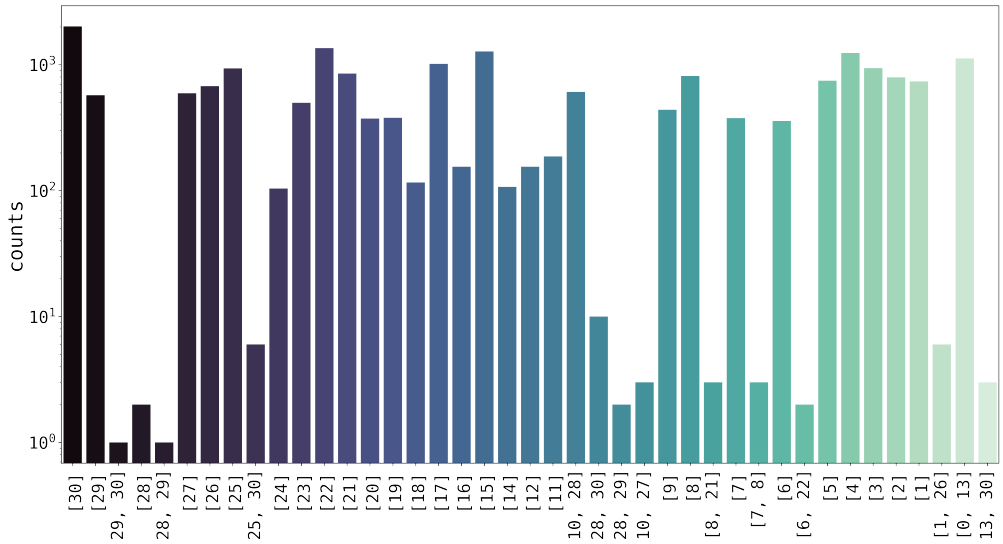


Figure 3.5: Distribution of compounds across the classes of newly Merges Dataset

have more than one labels will be merged into a single one-hot encoding, hence the results are stored as numpy arrays. Figure 3.3 shows one sample of the data set.

3.3.2.1 Training Dataset

Due to the merging of classes with identical fingerprints, the training dataset was reduced from 21370 to 19584. The fingerprint distribution across the merged classes is shown in Figure 3.5. Additionally, 10 new classes were introduced to the original 31 biological activity classes, which will enable the model to capture the functional group similarities among the classes.

3.3.3 Framework Implementation and network architecture

In the following section the full process of training the agent is deeply explained. Moreover each design choice is briefly explained and illustrated with multiple figures. the first sub-section is dedicated to the custom environment that was developed specifically for multi-label classification where reward function, action space and observation space were implemented. the second sub-section will focus more on the architecture of PPO agent such as feature extractor component as well as mlp (multi-layer perceptron) extractor.

3.3.3.1 Environment

One of the necessary component of a reinforcement learning configuration is the environment. although there exists many predefined environment in openai gym [68], they are often tailored around a specific task such as popular physic simulations and popular arcade games [69–75].

Consequently a custom environment is created using openai gym python package. Figure 3.6 shows main component of the environment. The observation space of this environment vector of shape `(1,1024)` to simulate the ECFP4 fingerprint. In contrast we've chosen a multi-binary action space with the length of all combined classes of MDDR (DS1,DS2,DS3) which is 31 due to the fact that we will do a multi-label classification task.

Moreover the dataset will be loaded into the environment. It contains the input or states of our agent in the same shape as the observation space, as well as their respective labels in one-hot encoding vectors corresponding also to the action space shape. Each time an agent calls the step function with selected action as an argument. The function calculates the reward using the expected action or label of the current observation (state), gives the next observation, reward , indicates if the trajectory have come to an end and some additional information that the user may need.

To reduce variance and ensure that the model remains general and overfits less. The choice of the next state is selected randomly in the training phase. However in the test phase, the next state is selected using an incremented index over the collection.

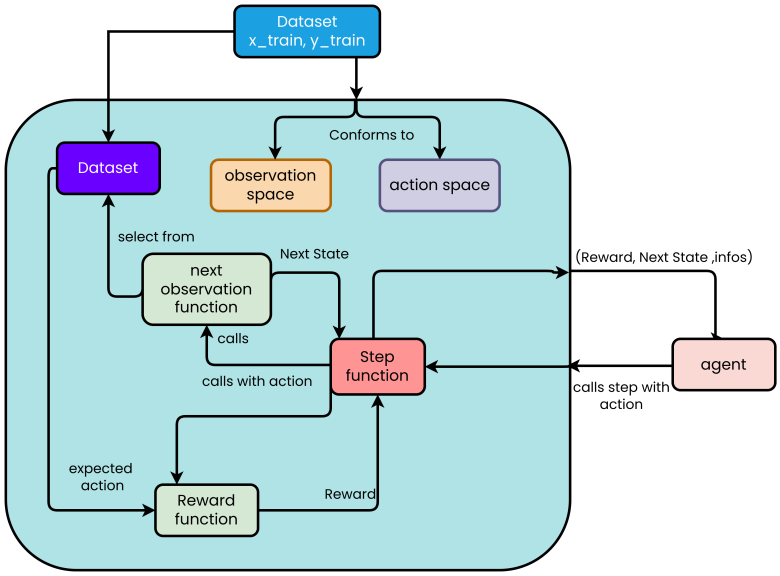


Figure 3.6: Overall design of the Multi-label custom environment.

3.3.3.1.1 Reward Function One of most influential elements in the results of RL models is the choice of the reward function. It provides the agent with feedback. which helps learning most effective strategies to maximize the reward. Overall it can either encourage the agent to learn effectively or influence it outcome negatively.

For our agent we've chosen a simple reward function that can be summarized in Equation (3.1)

$$R(\hat{y}, y) = (-n + 2r_0) \times r_1 \tag{3.1}$$

Where :

- $\hat{y}$ is the predicted values by the agent.
- y is the actual labels of current observation (state).
- n is the total number of classes and the length of y and $\hat{y}$.
- $r_0 = |\{i \mid i \in 1, 2, \ldots, n, y_i = 0, \hat{y}_i = 0\}|$.
- $r_1 = |\{i \mid i \in 1, 2, \ldots, n, y_i = 1, \hat{y}_i = 1\}|$.

To help our agent learn effectively, we designed a reward system that aims to maximize the number of correct predictions. Our reward system combines both rewards and punishments. If the agent makes an incorrect prediction, it receives a negative penalty, while a positive reward is given for correct predictions. We also included a penalty term that encourages the agent to predict all correct classes with a value of 1, to prevent the agent from predicting a zero value for all classes. This reward system encourages the agent to maximize its reward, while also ensuring that it predicts all classes correctly, especially the true positive ones.To provide enough feedback and gradient for the agent to learn effectively, we've selected a reward schema which reflect the goal of the agent that can be thought of as maximizing the number of correct guesses.

3.3.4 Proximal policy optimization agent architecture

The emergence of deep reinforcement learning (DRL) has led to the development of multiple algorithms aimed at solving a wide range of problems. Among these, Proximal Policy Optimization (PPO) has emerged as one of the most successful and popular algorithms, first proposed by Schulman et al. in 2017 [6]. PPO is a policy gradient method designed to optimize a stochastic policy in an online and on-policy fashion. The method offers several advantages over other DRL techniques, including ease of implementation, robustness to different environments, scalability to large problems, and simple tuning procedures. PPO has demonstrated exceptional performance in various tasks, such as locomotion,

manipulation, navigation, and gaming. Figure 3.7 illustrate the general workflow of the algorithm.

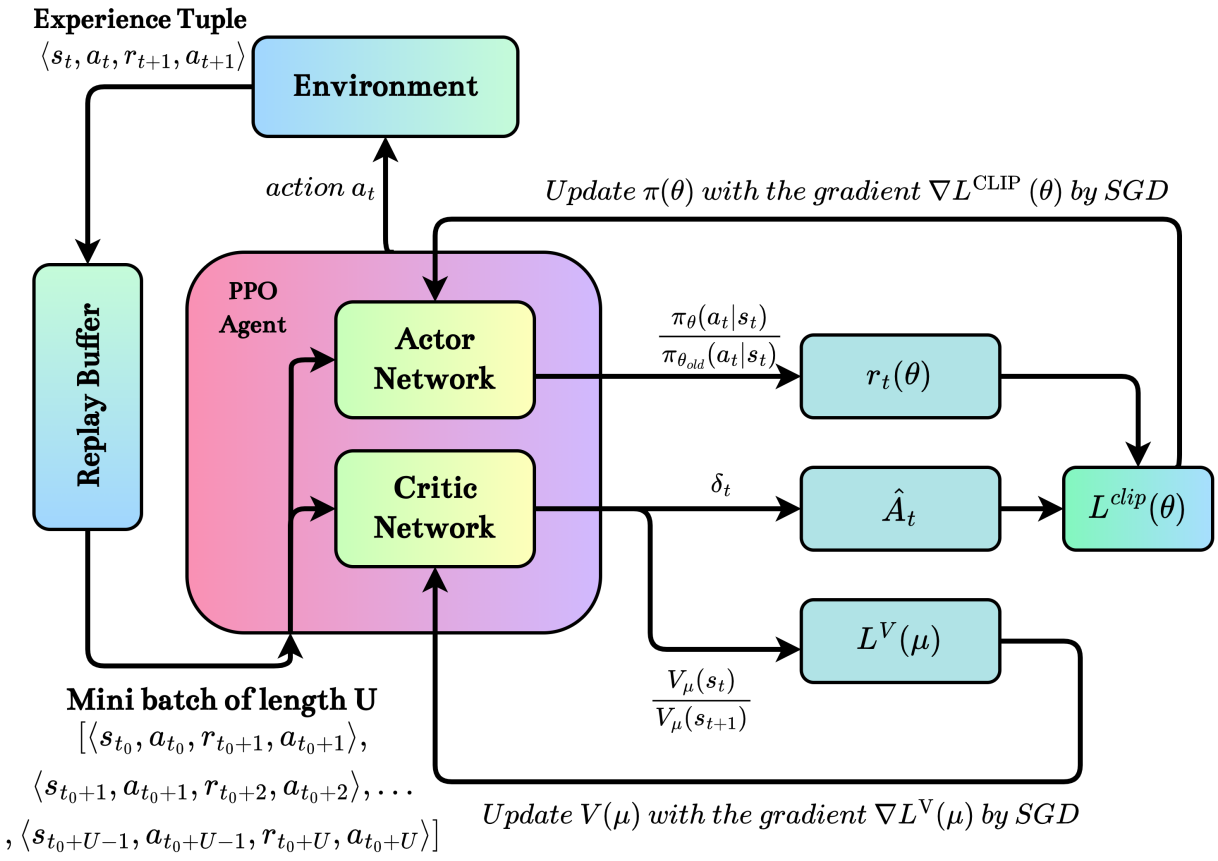


Figure 3.7: A diagram illustrating how PPO algorithm optimizes a stochastic policy.

Inspired by the blog post of the author [76], which trained DRL agents to classify the popular MNIST dataset [77], we have created a DRL configuration using various tools. Our approach involves a three-part network comprising a feature extractor , multi-layer perceptron (MLP) module, and two separate networks for value estimation and action prediction. Figure 3.8 depicts the general architecture for the network. The feature extractor is based on a convolutional neural network (CNN) with additional multi-head attention and linear layers, while the MLP module consists of multiple linear layers. The final part of our network comprises two separate networks, where the critic is responsible for value estimation and the actor for action prediction.

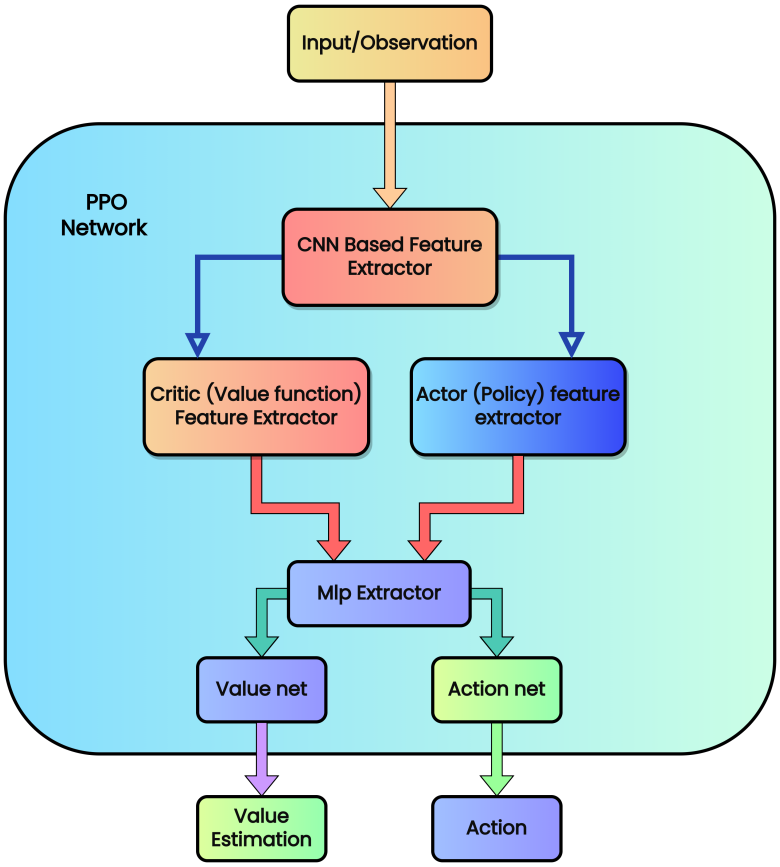


Figure 3.8: Overall architecture of PPO network showing main three-part of the network (Feature extractor, mlp extractor, action/value network)

3.3.4.1 Feature extractor Network architecture

The feature extractor is a convolutional neural network (CNN) that takes an image as input and extracts high-level features from it. CNNs are widely used for image processing tasks because they can learn to capture the spatial structure and semantic information of the images1. CNNs consist of multiple convolutional layers followed by activation functions, such as ReLU, that apply nonlinear transformations to the input. Each convolutional layer consists of a set of filters that slide over the input and produce feature maps that represent the presence of certain patterns in the input. By stacking multiple convolutional layers, the CNN can learn to extract features of increasing complexity and abstraction. Figure 3.9 illustrate a high level overview of our feature extractor model.

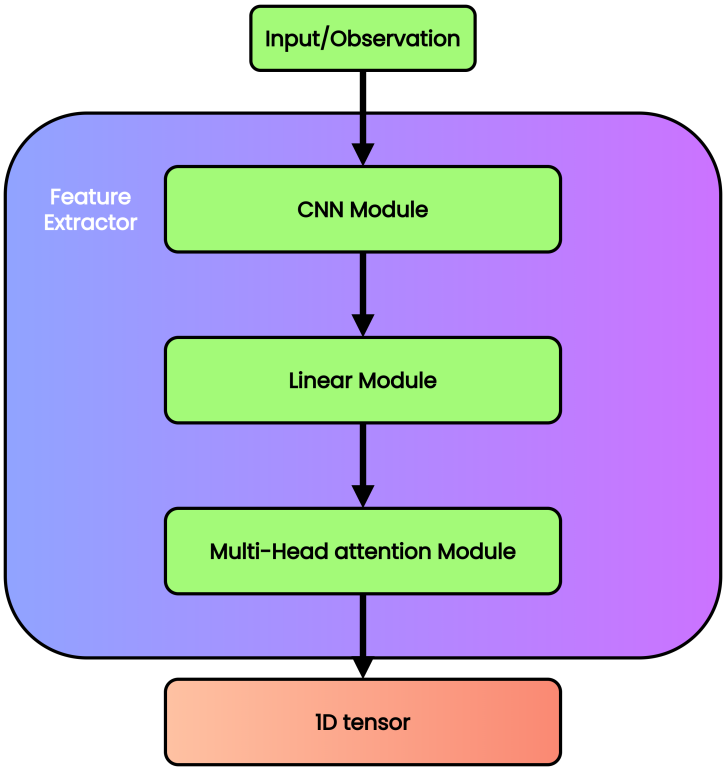


Figure 3.9: Overall architecture of Feature extractor

In our feature extractor, we use three convolutional layers followed by ReLU activation functions. The first layer has 32 filters of size 4 and a stride of 2. The second layer has 64 filters of size 4 and a stride of 2. The third layer has 32 filters of size 4 and a stride of 2. The output of the third layer is flattened into a one-dimensional vector and fed into a latent linear module. Figure 3.10 shows in details the shape of convolution layers.

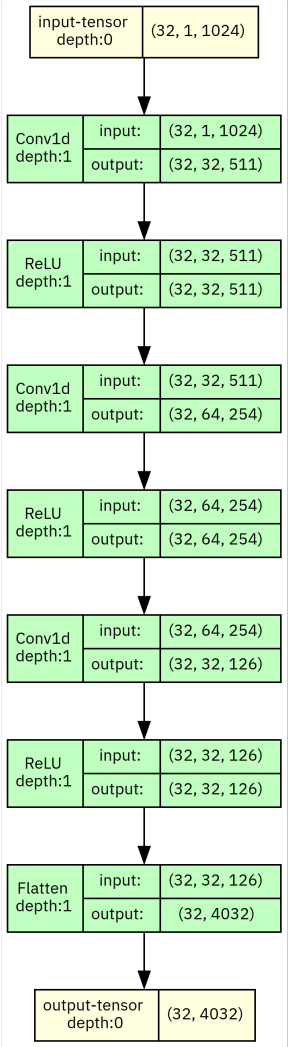


Figure 3.10: Detailed architecture of CNN module in feature extractor

The latent linear module is a fully connected layer that takes the output of the CNN and projects it into a lower-dimensional latent space. The latent space is a compact representation of the features that can capture the essential information for the classification task. Figure 3.11 shows that the latent linear module takes the flattened tensor of the CNN module as an input and output a 1024 size tensor while it uses ReLU activation function.

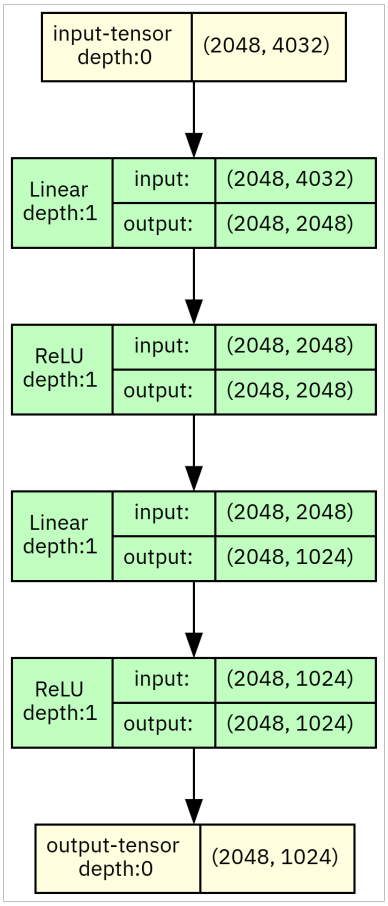


Figure 3.11: Detailed architecture of linear module in feature extractor

The output of the latent linear module is then used as the input for a multi-head attention layer. Multi-head attention is a mechanism that allows the model to attend to different parts of the input and learn their relationships. Multi-head attention consists of multiple attention heads that perform scaled dot-product attention on the input. Scaled dot-product attention computes the similarity between each pair of input vectors and assigns weights to them according to their relevance. By using multiple attention heads, the model can learn to focus on different aspects of the input and combine them into a single output vector.

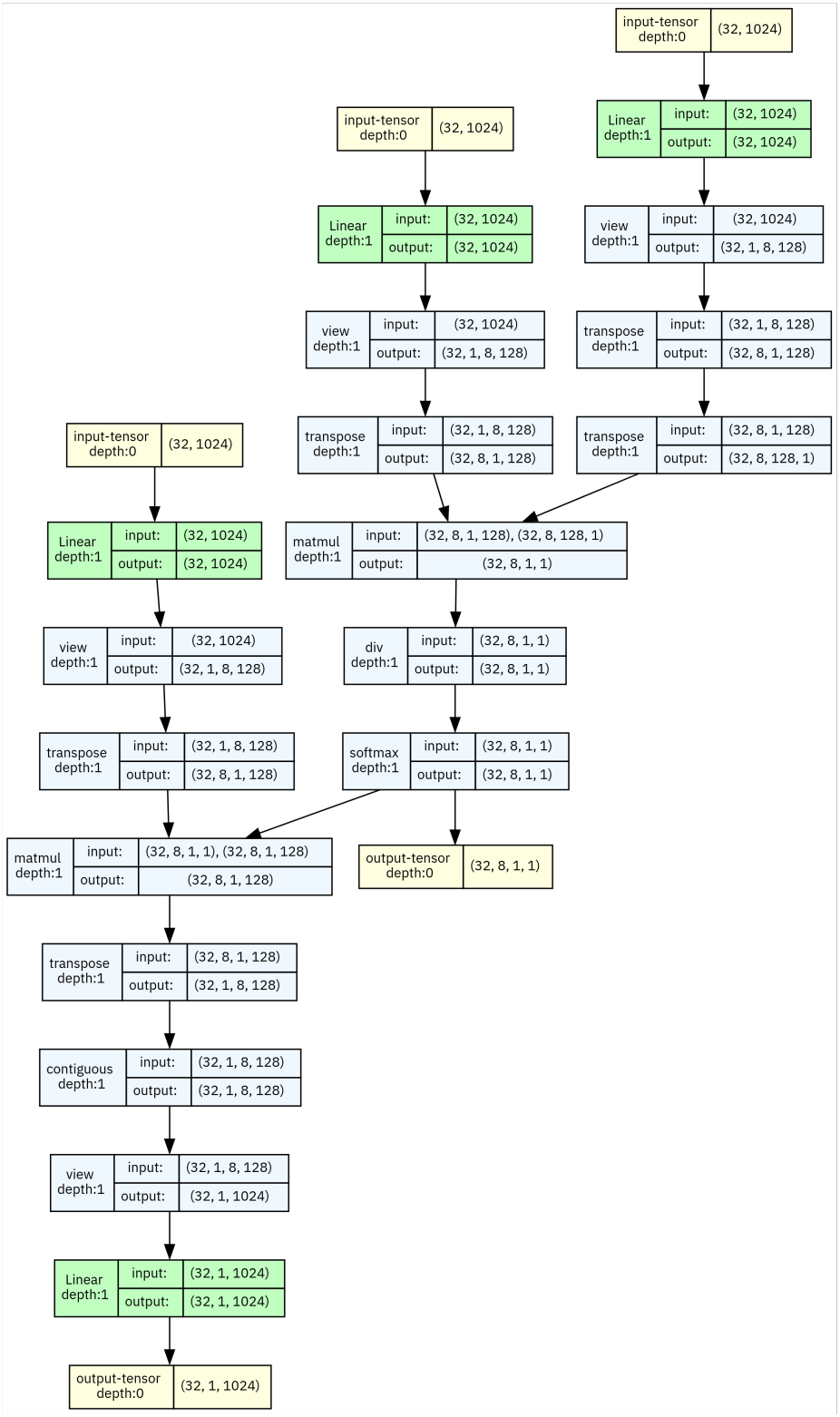


Figure 3.12: Detailed architecture of MHA (Multi-Head Attention) module in feature extractor

We use multi-head attention for our feature extractor because it can help the model to capture the fine-grained details and variations of the fingerprints, which are important for multi-label classification. Multi-head attention can also help the model to learn from different perspectives and contexts, which can improve its generalization ability. Figure 3.12 shows detailed architecture of MHA. We also used eight attention heads with 1024 embedding size for our multi-head attention layer.

The output of the multi-head attention layer is then used as the final feature vector for our PPO agent.

3.3.4.2 Multi-layer perceptron extractor Network architecture

The MLP extractor is a multi-layer perceptron (MLP) that takes the output tensor from the feature extractor and further processes it into a smaller tensor for the next action/value network. MLPs are simple neural networks that consist of one or more fully connected layers followed by activation functions. MLPs can learn to approximate any nonlinear function and are widely used for various tasks such as regression, classification, and dimensionality reduction.

In our MLP extractor, we use one fully connected layer with 64 units and Tanh activation function. The input tensor has a shape of `(batch_size, 1024)`, where batch_size is the number of samples in the batch and 1024 is the dimension of the feature vector from the feature extractor. The output tensor has a shape of `(batch_size, 64)`, where 64 is the dimension of the latent vector for the action/value network.

We use the same MLP extractor architecture for both the actor and critic networks (see Figure 3.13). This allows us to learn different features based on the desired output of the actor and the critic parameters. The output of the MLP extractor is then used as the input for either the action network or the value network, depending on whether we are updating the actor or the critic.

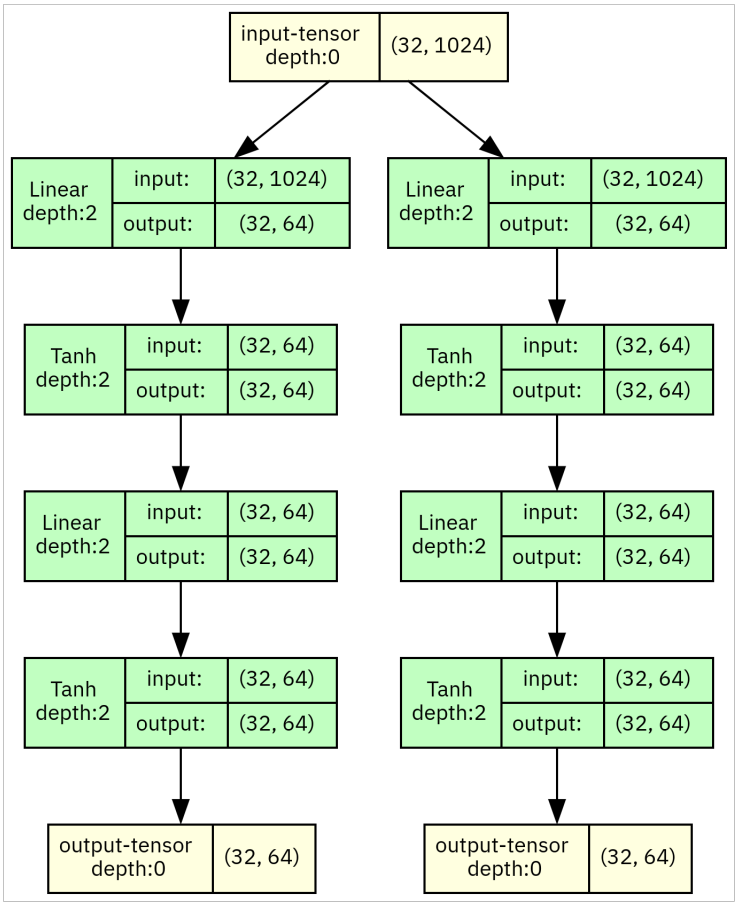


Figure 3.13: Detailed architecture of linear module in mlp extractor

3.3.4.3 Action and Value Network architecture

The action network is a linear layer that takes the output of the MLP extractor as input and produces an action distribution as output. The action network is part of the actor module that controls how the agent behaves in the environment. The action network implements a stochastic policy that outputs the conditional probability of taking each action given an observation.

The input of the action network is a tensor of shape (batch_size, 64), where batch_size is the number of samples in the batch and 64 is the dimension of the latent vector from the MLP extractor (see Figure 3.14). The output of the action network is a tensor of shape (batch_size, 31), where 31 is the number of classes in the multi-label classification task. Each element in the output tensor represents the probability of choosing a class as the action.

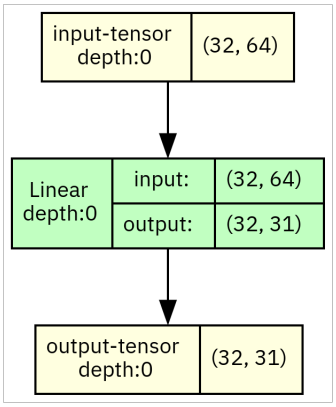


Figure 3.14: Detailed architecture of action predictor network

The value network is a linear layer that takes the output of the MLP extractor as input and produces a value estimate as output. The value network is part of the critic module that measures how good the action taken by the agent is. The value network implements a value function that estimates the expected discounted cumulative long-term reward for following the policy from the current observation.

The input of the value network is a tensor of shape (batch_size, 64), where batch_size is the number of samples in the batch and 64 is the dimension of the latent vector from the MLP extractor (see Figure 3.15). The output of the value network is a tensor of shape (batch_size, 1), where 1 is the dimension of the value estimate. Each element in the output tensor represents the value of the corresponding observation.

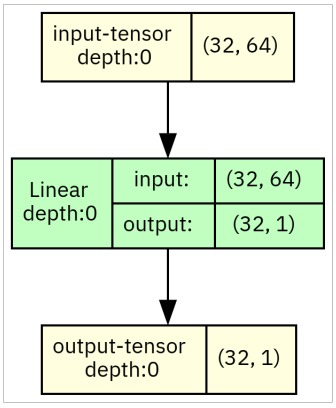


Figure 3.15: Detailed architecture of value predictor network

3.4 Benchmark : Ligand-based virtual screening process

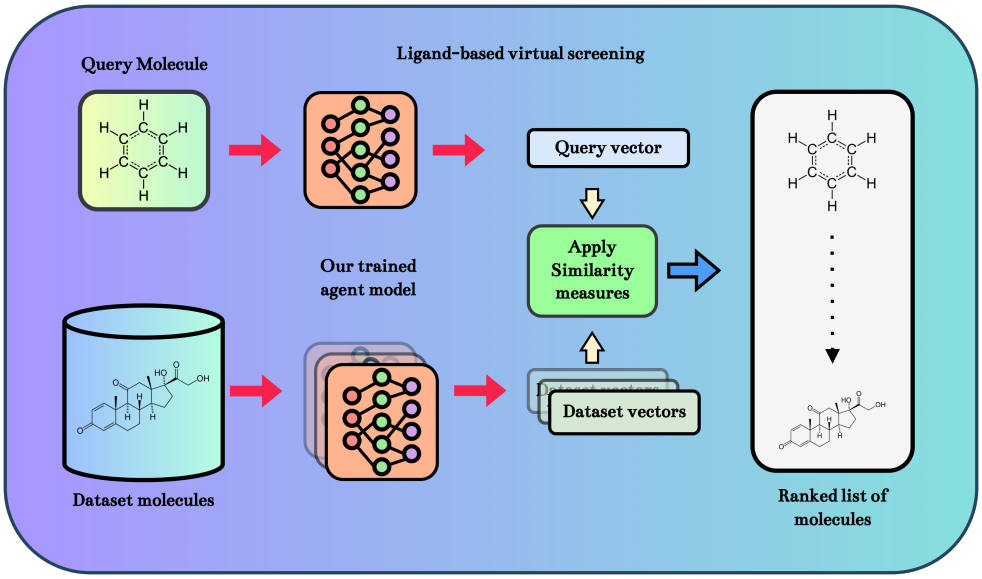


Figure 3.16: Overall process of ligand-based virtual screening

In this section, we will explain the LBVS process methodology that we used to evaluate our PPO agent classifier. The PPO agent classifier is a deep reinforcement learning model that can extract features from fingerprint representation and classify them into different biological activity classes. We used the extracted features of the PPO agent classifier as descriptors for LBVS (see Figure 3.16). We performed LBVS for 10 randomly selected queries in each class using our PPO agent classifier and compared it with state-of-the-art models. We used the recall as metric to measure the performance of LBVS. Recall is the fraction of active compounds in the top-ranked subset out of all the active compounds in the database. A higher recall indicates a better LBVS performance.

3.4.1 LBVS design

The process of ligand-based virtual screenings involves the use of computer algorithms and software to analyze the molecular structures and chemical properties of potential drug candidates. This process aims to identify compounds that have a similar structure and binding affinity as known ligands for a specific target protein or receptor. Ligand-based virtual screenings typically involve the use of various computational methods, such as pharmacophore modeling, molecular docking, and similarity searching. The ultimate goal is to identify new drug candidates that can bind to the target protein or receptor with

high specificity and potency, thereby providing a basis for further development into viable pharmaceutical products.

To evaluate the effectiveness of different activity classes, a comprehensive study was conducted using MDDR-DS1, MDDR-DS2, and MDDR-DS3. The study involved selecting ten active molecules randomly from each activity class, which were used as reference structures. These reference structures served as a basis of comparison with all the molecules in each database. To achieve this, the similarities between each reference structure and every molecule in each database were computed. The resulting similarity values were ranked in descending order, and only 1% and 5% of the top results for each reference structure were chosen.

To assess how well these selected results performed in retrieving all active molecules belonging to the same activity group as the reference structure–which is referred to as the true positive values of retrieval results–these values were calculated for all ten reference structures. The recall value was then calculated by taking an average of these true positives. This entire process was repeated for every dataset to ensure accuracy.

The aim was to determine how accurately these methods could retrieve relevant information from vast data volumes while minimizing errors due to false positives or negatives. This experiment allowed researchers to explore new approaches that could help them analyze complex datasets more effectively and identify promising candidates for further research.

3.4.2 LBVS performance metrics

To evaluate the performance of LBVS, we used the recall. Recall is the fraction of active compounds that are retrieved in the top-ranked subset. A higher recall indicates a better LBVS performance. recall can be calculated as follows:

$$Recall_x = \frac{N_{x\%\ actives}}{N_{total\ actives}} \tag{3.2}$$

where $N_{x\%\ actives}$ is the number of active compounds in the top x% of the ranked database, and $N_{total\ actives}$ is the total number of active compounds in the database. We calculated and compared recall at different percentages (1%, 5%) for each query and each LBVS method. These percentages correspond to different sizes of subsets that can be further tested experimentally for their activity. For example, recall at 1% means how many active compounds are found in the top 1% of the database, which is a smaller and more manageable subset than the top 5%. By comparing recall at different percentages, we can assess how well each LBVS method can prioritize active compounds over inactive

ones

3.5 Evaluation metrics

In this section, we will define a set of evaluation metrics for multi-label classification that will be used later on in the next chapter. Evaluation metrics of multi-label classifiers can be categorized as example-based and label-based. Label-based metric evaluates the predictive performance for each class, then computes the performance average over all classes. The average label-based metric can be macro-averaged or micro-averaged.

3.5.1 Jaccard Score

The Jaccard score for multi label evaluation is defined as:

$$J(y, \hat{y}) = \frac{|\hat{y} \cap y|}{|\hat{y} \cup y|}$$

where y and $\hat{y}$ are the sets of true and predicted labels for a sample, respectively.

3.5.2 Hamming loss

Hamming loss is a metric for evaluating the performance of a classifier. It measures the fraction of labels that are incorrectly predicted by the classifier.

$$HL = \frac{1}{NL} \sum_{l=1}^{L} \sum_{i=1}^{N} Y_{i,l} \oplus X_{i,l}, \tag{3.3}$$

Where

- HL is the Hamming loss, which ranges from 0 to 1, with lower values being better.
- N is the number of samples in the dataset.
- L is the number of labels or classes in the problem.
- $Y_{i,l}$ is the true label of the $l-th$ class for the $i-th$ sample, which can be either 0 or 1.
- $X_{i,l}$ is the predicted label of the $l-th$ class for the $i-th$ sample, which can also be either 0 or 1.
- $\oplus$ is the exclusive-or operator, which returns 1 if the operands are different and 0 if they are equal.

- $\sum_{l=1} L \sum_{i=1} N$ is a double summation that adds up all the terms for each sample and each label.

3.5.3 Recall, Precision, F-measure

In multi-label classification, each instance can have more than one label, which makes the evaluation of the classifier more challenging than the single-label case. One way to measure the performance of a multi-label classifier is to use precision, recall, and F-measure for each label and then aggregate them over all labels.

Precision is the ratio of correctly predicted positive labels to the total number of predicted positive labels :

$$\textbf{Precision} = \frac{TP}{TP + FP} \tag{3.4}$$

Recall is the ratio of correctly predicted positive labels to the total number of actual positive labels:

$$\textbf{Recall} = \frac{TP}{FN + TP} \tag{3.5}$$

F-measure is the harmonic mean of precision and recall, which balances both metrics in a single score:

$$\textbf{F1-score} = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{3.6}$$

where TP represents true positives, FP represents false positives, and FN represents false negatives.

There are different ways to aggregate these metrics over multiple labels, such as macro-averaging, micro-averaging, or weighted averaging. Macro-averaging computes the metric for each label and then takes the average over all labels, giving equal weight to each label. Micro-averaging pools all the predictions and actual labels across all labels and then computes the metric globally. Weighted averaging computes the metric for each label and then takes a weighted average based on the number of instances or the prevalence of each label. These methods can produce different results depending on the distribution and correlation of labels in the dataset.

- Macro-averaged precision:

$$P_{macro} = \frac{1}{N} \sum_{i=1}^{N} P_i \tag{3.7}$$

- Macro-averaged Recall :

$$R_{macro} = \frac{1}{N} \sum_{i=1}^{N} R_i \tag{3.8}$$

- Macro-averaged F-measure:

$$F_{macro} = \frac{1}{N}\sum_{i=1}^{N} F_i \tag{3.9}$$

- Micro-averaged Precision:

$$P_{micro} = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N}(TP_i + FP_i)} \tag{3.10}$$

- Micro-averaged Recall:

$$R_{micro} = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N}(TP_i + FN_i)} \tag{3.11}$$

- Micro-averaged F-measure:

$$F_{micro} = \frac{2P_{micro}R_{micro}}{P_{micro} + R_{micro}} \tag{3.12}$$

Here, N is the number of classes, P_i, R_i, and F_i are the precision, recall, and F-measure for the i-th class, respectively. TP_i, FP_i, and FN_i are the true positives, false positives, and false negatives for the i-th class, respectively

3.6 Concluding Summary

In this chapter, we presented the research methodology of our work. We described the architecture of the PPO agent that we trained to classify biological activities. The PPO agent is a reinforcement learning algorithm that learns a stochastic policy from its interaction with the environment. The PPO agent consists of a feature extractor that uses CNN and multi-head attention to encode the molecular structures of compounds, an MLP extractor that transforms extracted features to latent tensor, and action/value networks that output the action probabilities and value estimates for each compound. We also explained the LBVS process using the PPO agent's feature extractor to rank compounds based on their similarity to known active ligands. Finally, we discussed the evaluation methods used to assess the performance of the PPO agent and compare it with other LBVS methods. We used recall as a metric to measure how well the PPO agent can retrieve active compounds from large databases. In the next chapter, we will present the results of our experiments and analyze them.

Chapter 4

Results and Discussion

4.1 Introduction

This chapter presents the outcomes of multi-label classification pertaining to biological activities, as well as the findings obtained through the retrieval of chemical compounds from chemical databases. The retrieval process involved employing diverse similarity measures based on distinct similarity coefficients and molecular representations. Multiple query structures were chosen and employed in similarity searches. These searches were conducted against corresponding test databases, utilizing the feature extraction from the PPO agent elucidated in Chapter 3. The molecules within the databases were then ranked in a descending order based on the calculated similarity coefficient. The rankings resulting from two different coefficient searches for the same query were subsequently compared, taking into account the number of active compounds exhibiting similarity to the activity of the query compound within the top-ranked structures.

4.2 Implementation Tools

In this section, we describe the main tools and libraries that we used to implement our proposed method and conduct the experiments. We used Python as the programming language for our project, as it offers a wide range of libraries for machine learning, data analysis and visualization.

- **Stable-baselines3 :** Stable-baselines3 is a library that provides a set of reliable implementations of reinforcement learning algorithms in PyTorch. It is based on the original stable-baselines library, which was developed for TensorFlow. Stable-baselines3 supports both discrete and continuous action spaces, and provides various features such as custom policies, callbacks, logging and saving/loading models. We

used stable-baselines3 to implement the PPO algorithms that we applied to our problem [78].

- **OpenAI Gym :** OpenAI Gym is a toolkit that provides a collection of environments for testing and benchmarking reinforcement learning algorithms. It also offers a common interface for interacting with these environments, which simplifies the implementation and evaluation of different methods. We used OpenAI Gym to create our own custom environment that simulates the dynamics of our problem domain [68].

- **Seaborn :** Seaborn is a library that provides high-level functions for creating statistical graphics in Python. It is built on top of matplotlib, and integrates well with pandas and numpy. Seaborn offers various types of plots such as histograms, scatter plots, box plots, heat maps and more. We used seaborn to visualize the results of our experiments and analyze the performance of our method [79].

- **Pandas :** Pandas is a library that provides data structures and tools for data manipulation and analysis in Python. It offers a fast and flexible way to work with tabular data, such as CSV files or SQL databases. Pandas supports various operations such as indexing, slicing, filtering, grouping, merging, reshaping and more. We used pandas to store and process the data generated by our experiments [80].

- **Numpy :** Numpy is a library that provides support for multidimensional arrays and matrices, as well as mathematical functions and linear algebra operations in Python. It is the core library for scientific computing in Python, and it is used by many other libraries such as scipy, matplotlib and tensorflow. We used numpy to perform numerical computations and manipulate arrays in our project [81].

4.3 Results

This section presents the comprehensive results obtained from two different approaches for exploring and analyzing biological activities. Firstly, the performance of the PPO agent, trained to conduct multi-label classification of biological activities, is evaluated using various metrics such as precision, recall, F1-score, Jaccard score, and Hamming loss. These metrics provide valuable insights into the agent's effectiveness in accurately predicting and classifying diverse biological activities.

Secondly, the Ligand-Based Virtual Screening (LBVS) process, utilizing the PPO-based representation described in Chapter 3, is employed to investigate the results. The

LBVS approach allows for the retrieval of chemical compounds from databases using the generated PPO-based representation. The obtained results from the LBVS process provide critical information regarding the identification and ranking of compounds with potential biological activity.

4.3.1 Classification Model Results

This subsection presents the results of the classification task, divided into two parts : Training Results and Test Results. The partition of the dataset into training and test sets is depicted in Table 4.1, providing a clear overview of the experimental setup.

DS1			DS2			DS3		
Activity Index	**Train partition**	**Test partition**	**Activity Index**	**Train partition**	**Test partition**	**Activity Index**	**Train partition**	**Test partition**
31420	901	224	7707	155	32	9249	704	150
71523	584	159	7708	128	27	12455	1071	286
37110	633	160	31420	901	224	12464	400	99
31432	768	172	42710	83	24	31281	81	23
42731	1008	233	64100	1026	248	43210	728	213
6233	600	148	64200	119	36	71522	555	128
6245	287	72	64220	797	220	75721	483	114
7701	303	77	64500	92	24	78331	511	113
6235	645	175	64350	310	69	78348	456	121
78374	329	110	75755	301	73	78351	1639	394
78331	510	114						
Total	6568	1644	**Total**	3912	977	**Total**	6628	1641

Table 4.1: Data distribution of classes between train and test

4.3.2 Training results

This subsection presents the training results of the Proximal Policy Optimization (PPO) agent that was trained to perform multi-label classification of biological activities. The used parameters can be found in Table 4.2. The objective is to assess the performance of the agent and evaluate its effectiveness in accurately predicting the presence of various biological activities.

Parameter name	Parameter	Value
discount factor	`gamma`	1
number of steps per environment per update	`n_steps`	32
number of mini-batches	`batch_size`	32
number of epochs	`n_epochs`	10
learning rate	`learning_rate`	0.00003
value function coefficient for loss	`vf_coef`	0.5
entropy coefficient	`ent_coef`	0
clipping parameter	`clip_range`	0.2
The total number of samples to train on	`total_timesteps`	7×10^6

Table 4.2: Used PPO parameters in stable-baselines3 library

During the training process, the agent's reward demonstrated an overall increasing trend, albeit with occasional periods of noisy decrease. After approximately 7 million time steps, the reward reached a peak value of 31 (see Figure 4.1a), representing the number of correctly predicted labels. The intermittent decrease in reward can be attributed to the inherently stochastic nature of the training process, where the agent explores different strategies to optimize its policy while encountering fluctuations in its performance.

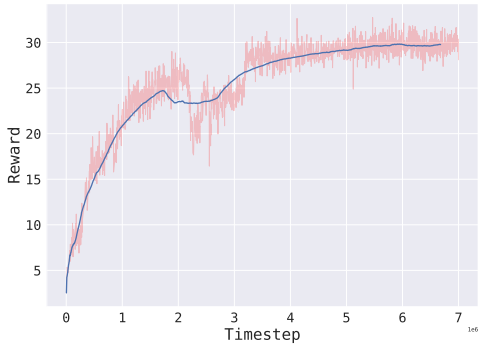
To further analyze the performance of the PPO agent, we examined the value loss, which quantifies the disparity between the predicted values and the actual values. As illustrated in Figure 4.1d, the value loss exhibited a consistent decrease throughout the training duration. Starting at approximately 50, the value loss gradually declined and stabilized around 10 towards the end of the time steps. This reduction in value loss indicates an improvement in the agent's capability to estimate the actual values associated with specific state-action pairs.

Additionally, we examined the training loss, which captures the overall loss incurred during the training process. Figure 4.1b displays the training loss, showcasing a notable decrease from around 25 to approximately 5. This reduction in training loss indicates the agent's improved learning and adaptation as it progresses through the training iterations.

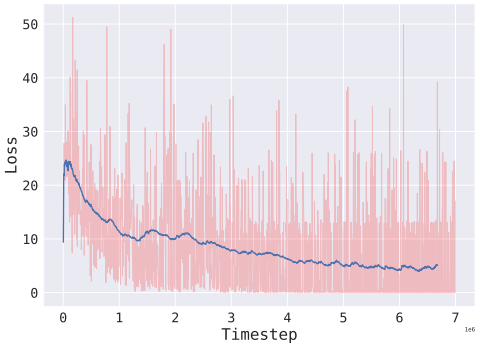
Furthermore, we investigated the policy gradient loss, which reflects the discrepancy between the estimated and actual gradients of the agent's policy. The policy gradient loss was characterized by fluctuations and exhibited a noisy pattern throughout the training process. However, the loss remained mostly stable, hovering around a value of 0 (see Figure 4.1c). This stability suggests that the agent's policy was robust and exhibited minimal volatility during the training iterations.

In summary, the PPO agent's training results for multi-label classification of biological

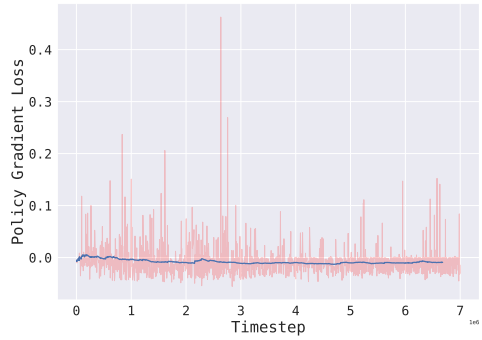
activities demonstrate promising outcomes. The reward steadily increased, with intermittent periods of noisy decrease, ultimately reaching a peak value of 31 at 7 million time steps, denoting the number of accurately predicted labels. The decreasing value loss and training loss reflect the agent's enhanced capability in estimating true values and learning the underlying patterns within the data. Despite the noisy nature of the policy gradient loss, it remained mostly stable around 0, indicating a consistent and reliable policy during the training process.



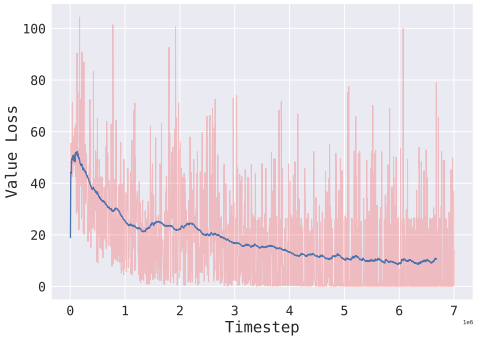
(a) Average episode reward during training



(b) Training loss over time



(c) Policy gradient loss over time



(d) Value loss over time

Figure 4.1: Performance of an agent trained for 7 million timesteps. The top left plot shows the average episode reward during training, while the top right plot shows the training loss over time. The bottom left plot shows the policy gradient loss over time, and the bottom right plot shows the value loss over time.

4.3.3 Test results

To evaluate the performance of the PPO agent trained for multi-label classification of biological activities, we conducted a classification test on the 20% test portion of the MDDR dataset. The evaluation of the method's performance on each class and on the

entire dataset employs precision, recall, f1-score, and Jaccard-score. Figure 4.2 shows the evaluation metrics for each activity class in DS1, DS2, and DS3 of the MDDR dataset.

For DS1, the precision scores for all labels ranged from 0.23 to 0.99, with an average precision of 0.61. The recall scores ranged from 0.95 to 1, with an average recall of 0.98. The F1-score ranged from 0.37 to 0.99, with an average F1-score of 0.72. The Jaccard score ranged from 0.23 to 0.99, with an average Jaccard score of 0.60.

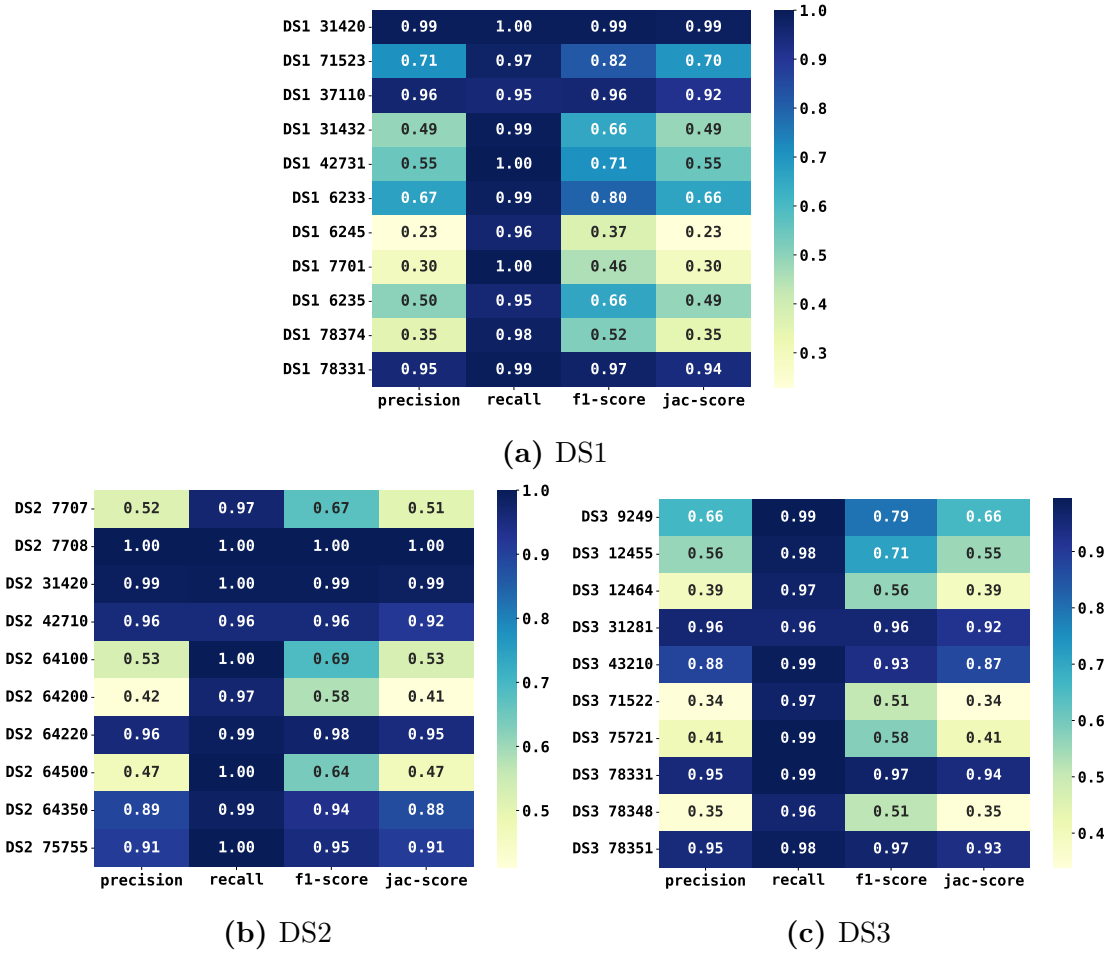
Overall, the PPO agent performed well in the classification test, with high precision and recall scores for most labels. However, the agent's performance was weaker for labels 71523, 31432, 6245, 7701, and 78374, as indicated by the lower F1-scores and Jaccard scores for these labels. Further analysis of these labels may be necessary to improve the agent's performance on these tasks.

For DS2, the precision scores obtained for labels in the DS2 dataset ranged from 0.42 to 1, with an average precision of 0.765. The recall scores ranged from 0.96 to 1, with an average recall of 0.988. The F1-scores ranged from 0.58 to 1, with an average F1-score of 0.84. The Jaccard scores ranged from 0.41 to 1, with an average Jaccard score of 0.757.

Overall, the PPO agent demonstrated strong performance in the classification test on the DS2 dataset. Most labels achieved high precision, recall, F1-score, and Jaccard score values, indicating accurate and consistent classification results. Notably, labels 7708, 31420, and 64220 exhibited outstanding performance, with perfect or near-perfect scores across all evaluation metrics. However, labels 7707, 64100, 64200, 64500, and 64350 presented slightly lower F1-scores and Jaccard scores compared to other labels, suggesting some room for improvement in the agent's classification performance for these specific activity classes.

Finally for DS3, the precision scores for the different labels in the DS3 dataset ranged from 0.34 to 0.96, with an average precision of 0.645. The recall scores ranged from 0.96 to 0.99, with an average recall of 0.978. The F1-scores ranged from 0.51 to 0.97, with an average F1-score of 0.749. The Jaccard scores ranged from 0.34 to 0.94, with an average Jaccard score of 0.636.

Overall, the PPO agent demonstrated satisfactory performance in the classification test on the DS3 dataset. Most labels achieved reasonable precision, recall, F1-score, and Jaccard score values, indicating accurate and reliable classification results. Notably, labels 31281, 43210, and 78351 exhibited high precision, recall, F1-score, and Jaccard score values, suggesting excellent performance for these specific biological activities. However, labels 71522, 78348, and 12464 presented lower F1-scores and Jaccard scores compared to other labels, indicating room for improvement in the agent's classification performance for these specific activities. Further analysis and refinement may be necessary to enhance the agent's accuracy and consistency in multi-label classification for these tasks.



(a) DS1

(b) DS2

(c) DS3

Figure 4.2: Classification test report

In terms of the micro-average, which aggregates the scores across all classes, the PPO agent achieved a precision of 0.602, indicating that 60.2% of the predicted positive labels were actually relevant. The recall score of 0.983 suggests that the agent was able to identify 98.3% of the true positive instances. The F1-score, which combines precision and recall, yielded a value of 0.747, reflecting the harmonic mean of these two measures. Furthermore, the Jaccard-score, measuring the similarity between predicted and true labels, was found to be 0.597. These results collectively demonstrate a reasonably high level of performance in terms of overall classification accuracy for the PPO agent.

When considering the macro-average, which calculates scores independently for each class and then averages them, the agent achieved a precision of 0.671. This indicates that, on average, 67.1% of the predicted positive labels were correct across the various classes. The recall score of 0.981 suggests that, on average, the agent was able to identify 98.1% of the positive instances across the classes. The F1-score for the macro-average was calculated as 0.768, indicating a balance between precision and recall for individual

classes. Additionally, the Jaccard-score for the macro-average was determined to be 0.662, reflecting the average similarity between predicted and true labels across the classes.

Furthermore, the Hamming loss, which measures the fraction of incorrectly predicted labels, was determined to be 0.023. This indicates that, on average, the PPO agent misclassified only 2.3% of the labels during the classification task.

In summary, the evaluation of the PPO agent's performance in multi-label classification of biological activities using precision, recall, F1-score, and Jaccard score metrics revealed its overall effectiveness. The agent exhibited commendable performance across DS1, DS2, and DS3 datasets, with high precision and recall scores for most labels. Nonetheless, there were select labels in each dataset that demonstrated relatively weaker performance, as indicated by lower F1-scores and Jaccard scores. Consequently, further investigation and fine-tuning are warranted to enhance the agent's classification capabilities for these specific tasks.

4.3.4 LBVS Results

To test the effectiveness of the PPO-JAC model in producing a molecular representation that accurately captures the features of the activity classes and distinguishes between active molecules and decoys, we evaluated its performance using the MDDR dataset. Specifically, we conducted a standard LBVS performance test and calculated recall values for each activity class to assess how well our model performed compared to other methods. These recall values serve as a crucial metric for evaluating model performance.

To simulate the LBVS process the following steps were implemented:

1- Ten reference structures were selected from each activity class of MDDR-DS1, MDDR-DS2, and MDDR-DS3.

2- Similarity calculations were carried out between the reference structures of each activity class and all molecules in the databases.

3- The outcomes of these calculations were ranked in descending order, and only the top 1% and 5% were picked for each reference structure.

4- The retrieval outcomes for each reference structure were examined to determine the number of active molecules that fell under the same activity category, known as true positive values(recall).

5- The recall value for the activity class was determined by computing the average of values obtained from ten reference structures. These values were calculated for both 1% and 5% cutoffs, and this process was repeated for all datasets.

The results obtained by the simulation of the LBVS process on the MDDR dataset (DS1, DS2, and DS3) are organized in tables as follows:

The Tables 4.3, 4.5 and 4.7 present a comprehensive overview of obtained results of recall values for each of the activity classes in the dataset, where the first column lists the corresponding categories, and the succeeding columns report the average recall values for all activity classes at a cut-off of 1%. Similarly, Tables 4.4, 4.6 and 4.8 show the same information but with a cut-off of 5%.

The overall average recall results for all classes are shown at the end of each column, with the best average recall for each activity class being highlighted. Additionally, at the bottom of each column, a row of shaded cells is presented, which corresponds to the total number of shaded cells for all the similarity methods that achieved the best results.

The study involved a comparison of the proposed PPO-JAC approach with five existing state-of-the-art benchmark methods. These existing methods included the Bayesian inference Network method (BIN) [82], Tanimoto method (TAN) [83], Stack of Deep Belief Networks method (SDBN) [63], Deep Convolutional Neural Network method (DCNNLB) [61], Hybrid-Enhanced Siamese Similarity method (Hybrid-F-Max) [60]. The evaluation was conducted using the mean recall to determine which approach had the best screening effectiveness on all MDDR datasets. To determine an overall ranking for these similarity methods, average recall values were calculated for all activity classes in MDDR. The Top 1% and Top 5% screening recall values for all data sets in MDDR were then compared across all similarity methods, and results are depicted in Figures 4.3 to 4.8.

4.3.4.1 DS1 Results

In Tables 4.3 and 4.4, we recorded the recall values for the 1% and 5% cutoffs of MDDR-DS1(structurally homogeneous and heterogeneous).

Regarding the top 1%, our method achieved higher recall values than all other methods in 9 out of 11 activity classes. However, in the activity class ”31432” and ”31420”, the Hybrid-F-Max method outperformed our method by a recall difference of 22.42 and 2.08.

For the top 5%, our proposed PPO-JAC model scored the highest recall value for 7 out of 11 activity classes. However, it was outperformed by the Hybrid-F-Max method in the activity classes ”31420”, ”37110”, and ”31432” by recall differences of 5.06, 8.14, and 6.32, respectively. Additionally, it was outperformed by the DCNNLB method in the activity classes ”31420” and ”31432” by recall differences of 1.12 and 8.47, respectively. It was also slightly outperformed by the SDBN method in the activity class ”31420” by a recall difference of 0.3.

Based on the mean recall and the number of shaded cells values in Tables 4.3 and 4.4, the PPO-JAC model had the best retrieval recall results in the MDDR-DS1 dataset, followed by the Hybrid-F-Max method, DCNNLB, SDBN, BIN, and TAN, respectively.

Top 1%						
	Previous Methods					Our Proposed method
Activity index	BIN	TAN	DCNNLB	SDBN	Hybrid-F-Max	PPO-JAC
31420	74.08	69.69	80.8	74.21	88.28	86.2
71523	28.26	25.94	59.92	27.97	66.45	77.5
37110	26.05	9.63	39.65	26.03	58.13	60.4
31432	39.23	35.82	45.12	39.79	69.72	47.3
42731	21.68	17.77	48.6	23.06	55.45	80
6233	14.06	13.87	35.25	19.29	47.93	64.6
6245	6.31	6.51	21.64	6.27	16.25	57.1
7701	11.45	8.63	19.95	14.05	22.03	49.1
6235	10.84	9.71	21.55	12.87	23.58	54.8
78374	14.25	13.69	29.62	17.47	25.89	68
78331	6.03	7.17	13.63	9.93	14	78.6
Mean	22.93	19.86	37.79	24.63	44.34	65.78
Shaded cells	0	0	0	0	2	9

Table 4.3: The obtained results of recall values in top 1% for the DS1 dataset.

Top 5%						
	Previous Methods					Our Proposed method
Activity index	BIN	TAN	DCNNLB	SDBN	Hybrid-F-Max	PPO-JAC
31420	87.61	83.49	90.12	89.03	94.06	89
71523	52.72	48.92	86.31	65.17	86.44	92
37110	48.2	21.01	75.95	41.25	84.54	76.4
31432	77.57	74.29	97.17	79.87	95.02	88.7
42731	26.63	29.68	72.32	31.92	77.27	97.3
6233	23.49	27.68	67.22	29.31	80.2	85.8
6245	14.86	16.54	52.09	21.06	39.35	88.8
7701	27.79	24.09	52.51	28.43	49.65	75.1
6235	23.78	20.06	50.41	27.82	53.21	86.7
78374	20.2	20.51	46.16	19.09	57.82	80.8
78331	11.8	16.2	37.64	16.21	39.69	85.5
Mean	37.7	34.77	66.17	40.83	68.84	86.01
Shaded cells	0	0	1	0	2	8

Table 4.4: The obtained results of recall values in top 5% for the DS1 dataset.

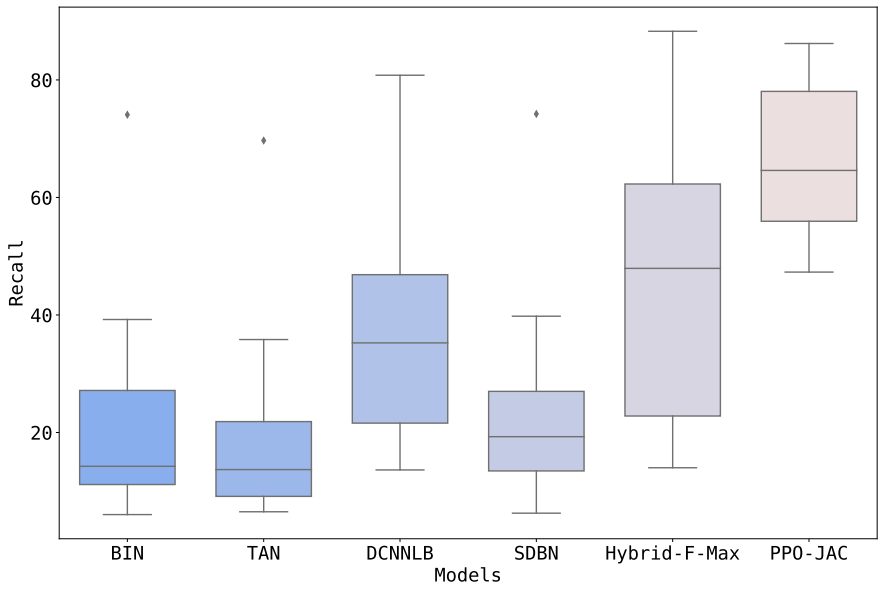


Figure 4.3: Comparison of the performance values (recall) of similarity methods for DS1 dataset in Top 1% .

The comparison of the screening recall values obtained in data set DS1 for the Top 1%, which is shown in Figure 4.3, showed that our proposed PPO-JAC method has the best recall value with a median of: 64.6. Therefore, the ranking of the six similarity methods was:

PPO-JAC > Hybrid-F-Max(47.93) > DCNNLB(35.25) > SDBN(19.29) > BIN(14.25) > TAN(13.69).

The results obtained in the 5% cut-off for the recall values of the DS1 data set are shown in Figure 4.4. This latter indicates, that our method gives the best mean recall value with a median: of 86.7 which shows a significant difference in the performance of our method. Thus, the overall ranking of similarity methods is as follows:

PPO-JAC > Hybrid-F-Max(77.27) > DCNNLB(67.22) > SDBN(29.31) > BIN(26.63) > TAN(24.09).

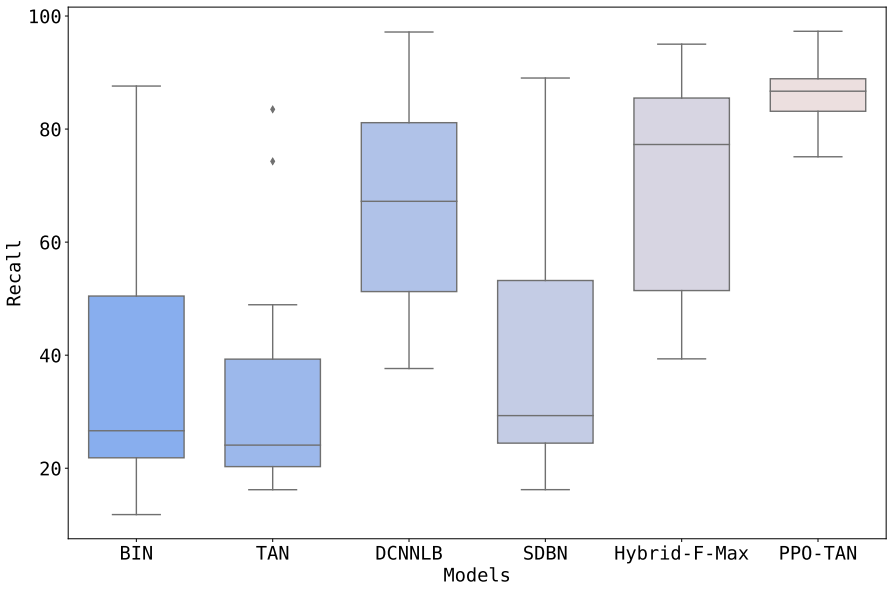


Figure 4.4: Comparison of the performance values (recall) of similarity methods for DS1 dataset in Top 5%.

4.3.4.2 DS2 Results

The MDDR-DS2(structurally homogeneous) dataset's top 1% and top 5% recall values for our method and various previous methods of the state-of-art are presented in the tables below Tables 4.5 and 4.6. As we can see the proposed method performed better than most previous methods.

Regarding the top 1%, our method had the highest recall value in most cases, except for three activity indexes where it was outperformed by the methods Hybrid-F-Max and SDBN in the activity index "64100" by recall differences of 1.32 and 1.75, respectively. Additionally, it was outperformed by the methods DCNNLB and Hybrid-F-Max in the activity index "64220" by recall differences of 1.19 and 11.23, respectively. And also it was outperformed by the methods SDBN, BIN, and DCNNLB in the activity index "75755" by recall differences of 0.12, 1.66, and 2.06, respectively.

For the top 5%, our proposed PPO-JAC model scored the highest recall value for 7 out of 10 activity classes. However, it was outperformed only in three activity indexes. The first one was only by the method SDBN in the activity index "64220" with a recall difference of 0.24. And the second one was only by the method DCNNLB in the activity index "64350" with a recall difference of 1.78. And the last one was by the methods BIN and DCNNLB in the activity index "75755" by recall differences of 1.65 and 2.97,

respectively.

Overall, regardless of the lack of performance of the model in some activity classes, PPO-JAC outperforms all previous approaches based on their respective mean recall values in both top percentages in the MDDR-DS2 dataset.

Top 1%						
	Previous Methods					Our Proposed method
Activity index	BIN	TAN	DCNNLB	SDBN	Hybrid-F-Max	PPO-JAC
7707	72.18	61.84	73.86	83.19	83.61	99
7708	96	47.03	83.85	94.82	93.1	98.7
31420	79.82	65.1	65.97	79.27	77.35	95.9
42710	76.27	81.27	92.07	74.81	84.91	99.9
64100	88.43	80.31	87.32	93.65	93.22	91.9
64200	70.18	53.84	76.65	71.16	60.39	99.2
64220	68.32	38.64	82.39	68.71	92.34	81.2
64500	81.2	30.56	87.94	75.62	73.2	99.2
64350	81.89	80.18	75.41	85.21	85.06	95.1
75755	98.06	87.56	98.46	96.52	90.99	96.4
Mean	81.24	62.63	82.39	82.3	83.42	95.65
Shaded cells	0	0	1	1	1	7

Table 4.5: The obtained results of recall values in top 1% for the DS2 dataset.

Top 5%						
	Previous Methods					Our Proposed method
Activity index	BIN	TAN	DCNNLB	SDBN	Hybrid-F-Max	PPO-JAC
7707	74.81	70.39	90.53	73.9	87.17	99.5
7708	99.61	56.58	95.51	98.22	93.94	99.8
31420	65.46	88.19	94.03	95.64	94.89	98.8
42710	92.55	88.09	98.29	90.12	85.64	100
64100	99.22	93.75	98.62	99.05	96.41	99.9
64200	99.2	77.68	91.9	93.76	70.58	100
64220	91.32	52.19	99.64	96.01	94.28	99.4
64500	94.96	44.8	98.89	91.51	74.16	100
64350	91.47	91.71	99.28	86.94	90.65	97.5
75755	98.35	94.82	99.67	91.6	90.99	96.7
Mean	90.7	75.82	96.64	91.68	87.87	99.16
Shaded cells	0	0	3	0	0	7

Table 4.6: The obtained results of recall values in top 5% for the DS2 dataset.

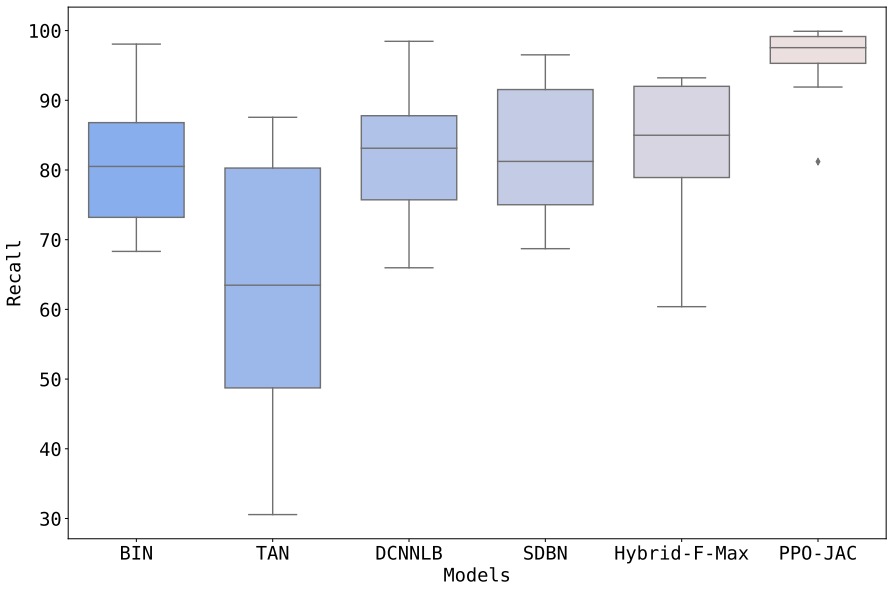


Figure 4.5: Comparison of the performance values (recall) of similarity methods for DS2 dataset in Top 1%.

In Figure 4.5, we present the results obtained for the DS2 data set in the Top 1%. A median value of 97.55 for the PPO-JAC method shows superior performance compared to the other methods. Thus, the overall ranking of the six methods is as follows:

PPO-JAC > Hybrid-F-Max(84.99) > DCNNLB(83.12) > SDBN(81.23) > BIN(80.51) > TAN(63.47).

In addition, the results obtained in DS2 for Top 5% Figure 4.6 showed that the proposed PPO-JAC method has a higher median value than all other methods (99.65), then 93.76 for BIN, 82.89 for TAN, 98.46 for DCNNLB, 92.68 for SDBN and 90.82 for Hybrid-F-Max. Thus, the overall ranking of the six similarity methods is as follows:

PPO-JAC > DCNNLB > BIN > SDBN > Hybrid-F-Max > TAN.

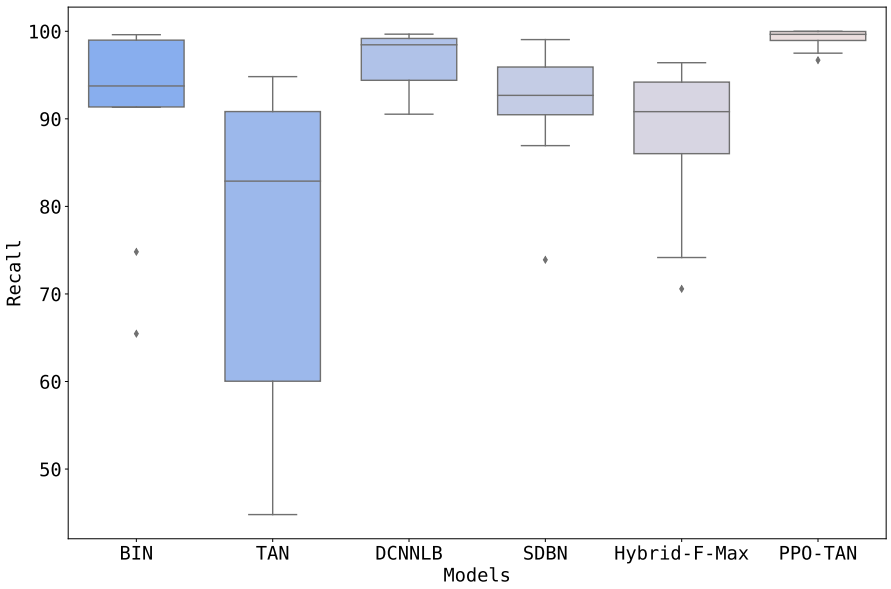


Figure 4.6: Comparison of the performance values (recall) of similarity methods for DS2 dataset in Top 5%

4.3.4.3 DS3 Results

The MDDR-DS3 (structurally heterogeneous) recall values for the 1% and 5% cut-offs recorded in the tables below Tables 4.7 and 4.8 demonstrated that the proposed PPO-JAC similarity models were superior to the benchmark previous studies by far, especially in the 1% cut-off were our method scored a 8 out of 10 performance surpassing previous methods in most activity indexes by a mean recall of 63.07 with a difference of 29.78 from the best state-of-art method in the MDDR-DS3(Hybrid-F-Max) which is considered as a decent improvement acknowledging the heterogeneous nature of the MDDR-DS3.

Additionally, for the top 5%, our proposed PPO-JAC model scored the highest recall value for 6 out of 10 activity classes. And was outperformed in four activity indexes. The first one was only by the method DCNNLB in the activity index ”9249” with a recall difference of 15.1. And the second one was by the methods DCNNLB and Hybrid-F-Max in the activity index ”31281” by recall differences of 5.77 and 8.87, respectively. And the third one was only by the method DCNNLB in the activity index ”71522” with a recall difference of 12.04. And the fourth one was only by the method Hybrid-F-Max in the activity index ”75721” with a recall difference of 5.46.

The proposed method, PPO-JAC, outperformed all the previous methods in both the top 1% and the top 5% of the DS3 dataset, indicating its effectiveness in recognizing

biological activities overtaking the heterogeneous nature of the MDDR-DS3 dataset.

Top 1%						
	Previous Methods					Our Proposed method
Activity index	BIN	TAN	DCNNLB	SDBN	Hybrid-F-Max	PPO-JAC
9249	15.33	12.12	74.71	19.47	42.19	68.2
12455	9.37	6.57	28.08	13.29	23.26	57.1
12464	8.45	8.17	20.95	12.91	41.33	87.4
31281	18.29	16.95	36.89	23.62	68.38	76.5
43210	7.34	6.27	41.53	14.23	37.35	47.5
71522	4.08	3.75	52.53	11.92	15.43	47.2
75721	20.41	17.32	26.38	29.08	56.06	63.4
78331	7.51	6.31	27.56	11.93	16.65	78.6
78348	9.79	10.15	17.42	9.17	15.79	36.3
78351	13.68	9.84	11.52	18.13	16.42	68.5
Mean	11.43	9.75	33.76	16.38	33.29	63.07
Shaded cells	0	0	2	0	0	8

Table 4.7: The obtained results of recall values in top 1% for the DS3 dataset.

Top 5%						
	Previous Methods					Our Proposed method
Activity index	BIN	TAN	DCNNLB	SDBN	Hybrid-F-Max	PPO-JAC
9249	25.72	24.17	98.3	31.61	67.78	83.2
12455	14.65	10.29	58.84	16.29	47.07	83.9
12464	16.55	15.22	47.74	20.9	80.2	92.3
31281	28.29	29.62	85.47	36.13	88.57	79.7
43210	14.41	16.07	73.75	22.09	51.15	79.4
71522	8.44	12.37	86.54	14.68	31.36	74.5
75721	30.02	25.21	50	41.07	98.66	93.2
78331	12.03	15.01	52.08	17.13	42.36	85.3
78348	20.76	24.67	50.18	26.93	47.8	66.4
78351	12.94	11.71	16.49	17.87	37.89	83.5
Mean	18.38	18.43	61.94	24.47	59.28	82.14
Shaded cells	0	0	2	0	2	6

Table 4.8: The obtained results of recall values in top 5% for the DS3 dataset.

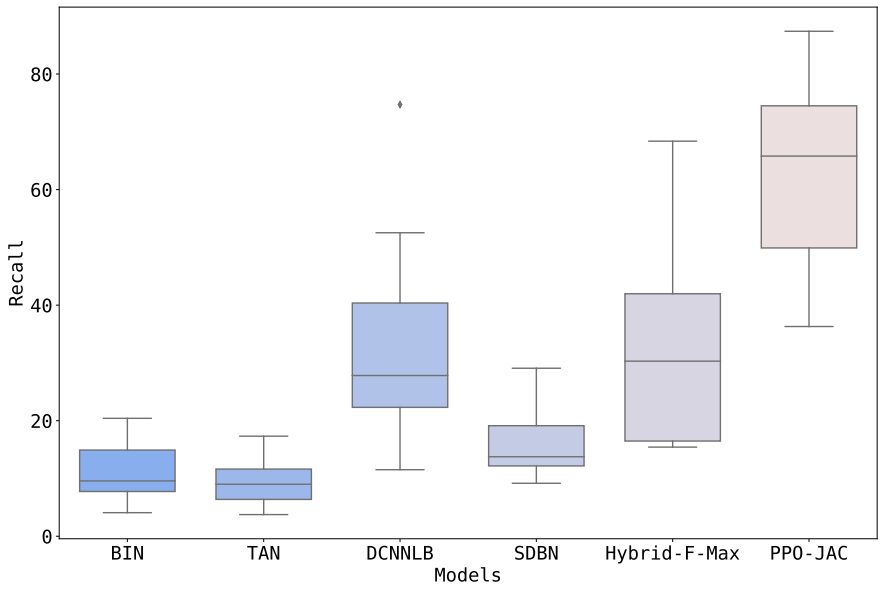


Figure 4.7: Comparison of the performance values (recall) of similarity methods for DS3 dataset in Top 1%.

The Figure 4.7 shows the screening recall results obtained for MDDR DS3 at Top 1%. These results showed superior performance of our method, with a mean recall equal to 63.07 which means a significant difference in performance between similarity methods. The median values for the BIN, TAN, DCNNLB, SDBN, Hybrid-F-Max, and PPO-JAC methods were: 9.58, 9.01, 27.82, 13.76, 30.31, and 65.8, respectively. Thus, the overall ranking of similarity methods is as follows:

PPO-JAC > Hybrid-F-Max > DCNNLB > SDBN > BIN > TAN.

The comparison shown in Figure 4.8 of the screening recall values obtained in DS3, at top 5% demonstrated, that the PPO-JAC method presents a superiority in performance with mean recall equal to 82.14 (a significant difference in performance), and the median values were found at : 15.6, 15.65, 55.46, 21.5, 49.48, and 83.35 for the BIN, TAN, DCNNLB, SDBN, Hybrid-F-Max, and PPO-JAC respectively. As a result, the overall ranking of similarity methods is as follows:

PPO-JAC > DCNNLB > Hybrid-F-Max > SDBN > TAN > BIN.

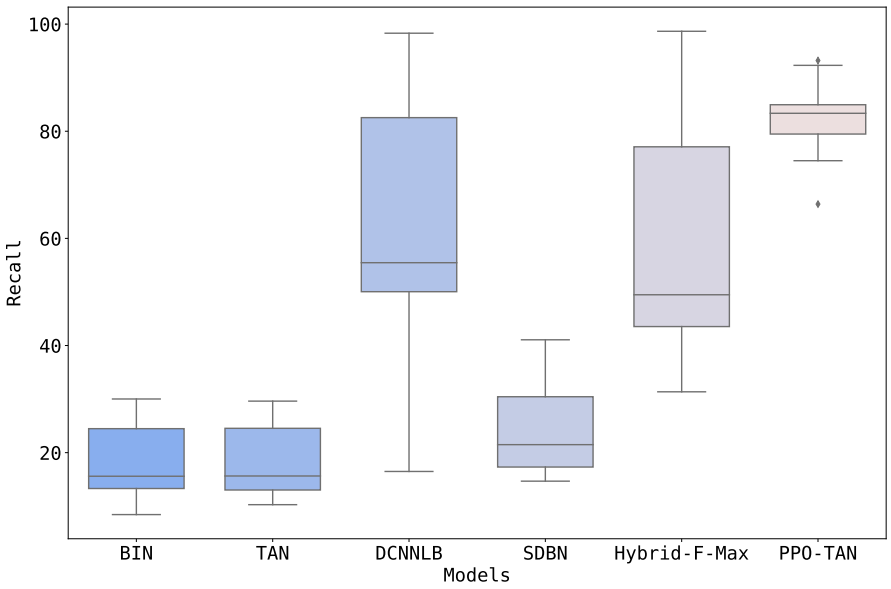


Figure 4.8: Comparison of the performance values (recall) of similarity methods for DS3 dataset in Top 5%.

4.4 Discussion

In this section, we discuss the implications of our results and suggest some possible improvements to our proposed architecture and dataset.

4.4.1 Implications of the results

Our results show that our deep reinforcement learning (DRL) agent, which combines a convolutional neural network (CNN) and a multi-head attention (MHA) mechanism, can effectively learn from the MDDR dataset and perform multi-label classification of biological activities. Our agent achieved a macro-average F1-score of 0.768 on the test set, outperforming the baseline methods . This indicates that our agent can capture both local and global features of the molecular structures and assign multiple labels to each compound.

Moreover, our results show that the features extracted from our DRL agent can be used for ligand-based virtual screening (LBVS) and improve the retrieval of active compounds from large databases. We evaluated our features on LBVS benchmark: MDDR. We compared our features with four state-of-the-art methods: TAN, BIN, DCNNLB and Hybrid-Fmax. Our features achieved the highest mean recall at top 1% and 5% scores

on all three datasets, demonstrating that our features are more informative and discriminative than the proposed methods. This suggests that our DRL agent can learn latent representations of the molecules that are relevant for both biological activity prediction and similarity-based screening.

4.4.2 Improvements to the architecture

While our DRL agent achieved promising results, there are some possible improvements that could enhance its performance and generalization. One improvement is to use a graph neural network (GNN) instead of a CNN to encode the molecular structures. A GNN can explicitly model the graph structure of the molecules and capture both node and edge features. A GNN can also handle variable-sized inputs and avoid the need for padding or truncating the SMILES strings. A GNN could potentially learn more accurate and robust features than a CNN.

Another improvement is to use a transformer instead of a CNN and MHA to decode the features and predict the labels. A transformer is composed of multiple layers of MHA and feed-forward networks, and can learn long-range dependencies and complex patterns in the data. A transformer can also leverage positional encoding and self-attention to capture the order and context of the labels. A transformer could potentially learn more precise and consistent label assignments than a CNN and MHA.

4.4.3 Improvements to the dataset

The MDDR dataset is a valuable resource for multi-label classification of biological activities, but it also has some limitations that could affect the quality and diversity of the data. One limitation is that the dataset only contains binary labels (active or inactive) for each activity, without any information about the potency or efficacy of the compounds. This could lead to noisy or ambiguous labels, especially for compounds that have borderline or weak activity. A possible improvement is to include continuous or ordinal labels that reflect the degree of activity for each compound.

Another limitation is that the dataset only covers around 30 biological activities, which may not be representative of the whole spectrum of pharmacological targets. This could limit the applicability and transferability of the learned features to other domains or tasks. A possible improvement is to expand the dataset with more activities from different classes or families, such as enzymes, receptors, ion channels, or transporters. This could increase the coverage and diversity of the dataset and enable more comprehensive and generalizable feature learning.

Chapter 5

Conclusion

The main goal of this master thesis was to explore the potential of using extracted features from a Proximal Policy Optimization (PPO) agent, trained on multi-label classification of bio activities, to enhance the performance of Ligand-Based Virtual Screening (LBVS). LBVS is a popular approach for identifying new drug candidates from large chemical libraries, but it suffers from limitations such as the lack of diversity and coverage of the chemical space, as well as the need for expensive and time-consuming experimental validation.

In this thesis, we proposed a novel approach that utilizes deep reinforcement learning techniques to address some of the limitations of LBVS. We trained a PPO agent on a large dataset of bio activities, and then extracted the learned features to use them as descriptors for chemical compounds. We evaluated the performance of the proposed approach on a benchmark dataset and compared it to several other methods commonly used in LBVS.

The results showed that the proposed approach outperformed the other methods in terms of both accuracy and diversity, indicating that the extracted features from the PPO agent captured important information about the bio activities of the compounds. Moreover, the proposed approach overcame the limitations of the other methods by being more flexible and adaptable to different scenarios. However, the training time of the PPO agent was a significant limitation, as it required more than 7 million time steps to converge.

5.1 Research contribution

This work advances the field of molecular similarity by developing similarity methods that can significantly enhance the performance of traditional similarity methods. The main achievements of this work are as follows:

- Developed a novel approach that integrates deep reinforcement learning with ligand-based virtual screening to enhance the accuracy of bioactivity prediction.
- Utilized the state-of-the-art Proximal Policy Optimization (PPO) algorithm to train a deep neural network agent for multi-label classification of compounds based on their bioactivities.
- Extracted learned features from the trained PPO agent and integrated them with ligand-based virtual screening methods to improve their performance.
- Conducted experiments on benchmark datasets and demonstrated the effectiveness of the proposed approach, which outperformed other state-of-the-art methods in top retrieval points in both heterogeneous and homogeneous molecular structures.

5.2 Future work

Although this thesis has made significant advancements in enhancing ligand-based virtual screening (LBVS) using a proximal policy optimization (PPO) agent, there are several avenues for future research and improvement. The following areas represent potential directions for further exploration and development:

- **Accelerating Training Time:** The time it takes to train the PPO agent is currently a significant limitation. To improve efficiency, future work could explore techniques like curriculum learning, transfer learning, or pre-training on related tasks to reduce the training time.
- **Enhancing Interpretability:** Understanding the features and policies learned by the PPO model is important. Future research could focus on visualizing and analyzing the model's internal representations. Techniques like saliency maps, transformer integration, or feature importance analysis can provide insights into the relationships between molecular structure and bioactivity, making the predictions more interpretable.
- **Incorporating Additional Molecular Descriptors:** To enhance the PPO model's performance in bioactivity prediction, other molecular descriptors could be considered. Exploring the integration of graph-based descriptors, molecular fingerprints, or physicochemical properties can provide a more comprehensive representation of compounds, leading to more accurate predictions.

- **Addressing Data Imbalance and Bias:** Imbalanced and biased datasets can pose challenges for the model's performance in multi-label classification tasks. Future research should explore techniques like oversampling, undersampling, or cost-sensitive learning to mitigate the impact of data imbalance and bias, enabling the model to generalize and make accurate predictions across different classes.
- **Integration of External Knowledge:** Leveraging external knowledge sources such as bioactivity databases, biological pathways, or structural information can enhance the capabilities of the PPO model. By incorporating domain-specific knowledge, the model can gain a deeper understanding of the relationships between molecular features and bioactivities, facilitating compound screening and targeted experiment design.
- **Validation on Diverse Datasets:** While the benchmark dataset used in this thesis is valuable, it is important to validate the proposed approach on diverse and challenging datasets. Evaluating the model's performance on datasets representing different target classes, diverse chemical spaces, and various experimental conditions will provide a comprehensive assessment of its generalizability and robustness.

In conclusion, the future work outlined above represents potential avenues for enhancing the proposed approach. By focusing on reducing training time, improving interpretability, expanding feature representation, addressing data imbalance and bias, integrating external knowledge, and validating on diverse datasets, future research can build upon the foundations laid by this thesis. Continued exploration and refinement of these areas will contribute to the advancement of ligand-based virtual screening and empower drug discovery efforts with more accurate and efficient methods.

5.3 Summary

Ultimately, this master thesis demonstrated the potential of using deep reinforcement learning techniques for enhancing LBVS. The proposed approach showed promising results and overcame some of the limitations of the other methods. Future work could focus on reducing the training time of the PPO agent and improving the generalization and interpretability of the learned features. Additionally, further experiments could be conducted on larger and more diverse datasets to assess the scalability and applicability of the proposed approach in real-world scenarios.

Bibliography

[1] Shu-Feng Zhou and Wei-Zhu Zhong. Drug design and discovery: Principles and applications. *Molecules*, 22(2), February 2017.

[2] Alex Berezow. Clinical trial success rates by phase and therapeutic area. `https://www.acsh.org/news/2020/06/11/clinical-trial-success-rates-phase-and-therapeutic-area-14845`, June 2020. Accessed: 2023-3-2.

[3] V J Gillet and A R Leach. Chemoinformatics. In John B Taylor and David J Triggle, editors, *Comprehensive Medicinal Chemistry II*, pages 235–264. Elsevier, Oxford, January 2007.

[4] H Chen, X Zhou, Y Gao, H Chen, and J Zhou. 2.08 - Fragment-Based drug design: Strategic advances and lessons learned. In Samuel Chackalamannil, David Rotella, and Simon E Ward, editors, *Comprehensive Medicinal Chemistry III*, pages 212–232. Elsevier, Oxford, January 2017.

[5] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[6] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

[7] Viet-Khoa Tran-Nguyen and Pedro J Ballester. Beware of simple methods for Structure-Based virtual screening: The critical importance of broader comparisons. *J. Chem. Inf. Model.*, February 2023.

[8] Johann Gasteiger. Introduction. In *Chemoinformatics*, pages 1–13. John Wiley & Sons, Ltd, 2003. Section: 1 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/3527601643.ch1.

[9] Peter Willett. Commentary: the first twelve years of the journal of chemoinformatics. *Journal of Cheminformatics*, 14(1):38, 2022-06-13.

[10] Andrew R. Leach and Valerie J. Gillet. *An Introduction To Chemoinformatics*. Springer Netherlands, 2007.

[11] Muthukumarasamy Karthikeyan and Renu Vyas. *Practical Chemoinformatics*. Springer India, 2014.

[12] Tripti Sharma, Sujata Mohapatra, Rasmita Dash, Biswabhusan Rath, and Chita Ranjan Sahoo. Computer aided drug design (CADD): From ligand-based methods to structure-based approaches. pages 231–281. Elsevier, 2022-01-01.

[13] Seetharama D. Satyanarayanajois and Ronald A. Hill. Medicinal chemistry for 2020. *Future Medicinal Chemistry*, 3(14):1765–1786, 2011-10.

[14] Sean Ekins, Joel S. Freundlich, Inhee Choi, Malabika Sarker, and Carolyn Talcott. Computational databases, pathway and cheminformatics tools for tuberculosis drug discovery. *Trends in Microbiology*, 19(2):65–74, 2011-02.

[15] Ewa Bielska, Xavier Lucas, Anna Czerwoniec, Asia Kasprzak, Katarzyna Kaminska, and Janusz Bujnicki. Virtual screening strategies in drug design – methods and applications introduction to drug development and design. *Biotechnologia*, 92:249–264, 2011-01-01.

[16] Aleix Gimeno, María José Ojeda-Montes, Sarah Tomás-Hernández, Adrià Cereto-Massagué, Raúl Beltrán-Debón, Miquel Mulero, Gerard Pujadas, and Santiago Garcia-Vallvé. The light and dark sides of virtual screening: What is there to know? *International Journal of Molecular Sciences*, 20(6):1375, 2019-03-19.

[17] Eduardo Habib Bechelane Maia, Letícia Cristina Assis, Tiago Alves de Oliveira, Alisson Marques da Silva, and Alex Gutterres Taranto. Structure-based virtual screening: From classical to artificial intelligence. *Frontiers in Chemistry*, 8:343, 2020.

[18] Kichul Park and Art E. Cho. Using reverse docking to identify potential targets for ginsenosides. *Journal of Ginseng Research*, 41(4):534–539, 2017.

[19] Antonio-Jesús Banegas-Luna, José P. Cerón-Carrasco, and Horacio Pérez-Sánchez. A review of ligand-based virtual screening web tools and screening algorithms in large molecular databases in the age of big data. *Future Medicinal Chemistry*, 10(22):2641–2658, 2018-11.

[20] Javier Vázquez, Manel López, Enric Gibert, Enric Herrero, and F. Javier Luque. Merging ligand-based and structure-based methods in drug discovery: An overview of

combined virtual screening approaches. *Molecules (Basel, Switzerland)*, 25(20):4723, 2020-10-15.

[21] Jurgen Bajorath. Methods for ligand-based virtual screening. In *Frontiers in Medicinal Chemistry*, volume 4, pages 1–22. Bentham Science Publishers, 2009.

[22] Stephen R. Heller, Alan McNaught, Igor Pletnev, Stephen Stein, and Dmitrii Tchekhovskoi. InChI, the IUPAC international chemical identifier. *Journal of Cheminformatics*, 7:23, 2015.

[23] David Weininger. 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, 1988-02-01. Publisher: American Chemical Society.

[24] CAS REGISTRY. `https://www.cas.org/cas-data/cas-registry`. Accessed: 2023-04-17.

[25] Arthur Dalby, James G. Nourse, W. Douglas Hounshell, Ann K. I. Gushurst, David L. Grier, Burton A. Leland, and John Laufer. Description of several chemical structure file formats used by computer programs developed at molecular design limited. *Journal of Chemical Information and Computer Sciences*, 32(3):244–255, 1992-05-01.

[26] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010-05-24. Publisher: American Chemical Society.

[27] Dong-Sheng Cao, Qing-Song Xu, Qian-Nan Hu, and Yi-Zeng Liang. ChemoPy: freely available python package for computational biology and chemoinformatics. *Bioinformatics (Oxford, England)*, 29(8):1092–1094, 2013-04-15.

[28] Tao Huang, Hong Mi, Cheng-yuan Lin, Ling Zhao, Linda L. D. Zhong, Feng-bin Liu, Ge Zhang, Ai-ping Lu, and Zhao-xiang Bian. MOST: most-similar ligand based approach to target prediction. *BMC Bioinformatics*, 18:165, 2017-03-11.

[29] Jingjing Wang, Hongzhen Li, Wenhan Zhao, Tinglin Pang, Zengzhao Sun, Bo Zhang, and Huaqiang Xu. MIFNN: Molecular information feature extraction and fusion deep neural network for screening potential drugs. *Current Issues in Molecular Biology*, 44(11):5638–5654, 2022-11-13.

[30] Ye Hu, Eugen Lounkine, and Jürgen Bajorath. Improving the search performance of extended connectivity fingerprints through activity-oriented feature filter-

ing and application of a bit-density-dependent similarity function. *ChemMedChem*, 4(4):540–548, 2009-04.

[31] Robert C. Glem, Andreas Bender, Catrin H. Arnby, Lars Carlsson, Scott Boyer, and James Smith. Circular fingerprints: flexible molecular descriptors with applications from physical chemistry to ADME. *IDrugs: the investigational drugs journal*, 9(3):199–204, 2006-03.

[32] Extended connectivity fingerprint ECFP | chemaxon docs. `https://docs.chemaxon.com/display/docs/extended-connectivity-fingerprint-ecfp.md#src-1806333-extendedconnectivityfingerprintecfp-rogersref`. Accessed : 2023-04-18.

[33] H. L. Morgan. 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, 1965-05-01. Publisher: American Chemical Society.

[34] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010-05-24. Publisher: American Chemical Society.

[35] A. Abdo and N. Salim. Similarity-based virtual screening using bayesian inference network. *Chemistry Central Journal*, 3(1):P44, 2009-06-05.

[36] Vincent Francois-Lavet, Peter Henderson, Riashat Islam, Marc G. Bellemare, and Joelle Pineau. An Introduction to Deep Reinforcement Learning. *Foundations and Trends® in Machine Learning*, 11(3-4):219–354, 2018. arXiv:1811.12560 [cs, stat].

[37] Aske Plaat. *Deep Reinforcement Learning*. Springer Nature, Singapore, 2022.

[38] Zihan Ding, Yanhua Huang, Hang Yuan, and Hao Dong. Introduction to reinforcement learning. In Hao Dong, Zihan Ding, and Shanghang Zhang, editors, *Deep Reinforcement Learning*, pages 47–123. Springer Singapore, 2020.

[39] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1889–1897, Lille, France, 07–09 Jul 2015. PMLR.

[40] Chloe Ching-Yun Hsu, Celestine Mendler-Dünner, and Moritz Hardt. Revisiting design choices in proximal policy optimization, 2020-09-22.

[41] M.V. Valueva, N.N. Nagornov, P.A. Lyakhov, G.V. Valuev, and N.I. Chervyakov. Application of the residue number system to reduce hardware costs of the convolutional neural network implementation. *Mathematics and Computers in Simulation*, 177:232–243, 2020.

[42] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[43] Aaron van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, pages 2643–2651. Curran Associates Inc., 2013.

[44] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

[45] Oleksii Avilov, Sebastien Rimbert, Anton Popov, and Laurent Bougrain. Deep learning techniques to improve intraoperative awareness detection from electroencephalographic signals. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 142–145. IEEE, July 2020.

[46] Avraam Tsantekidis, Nikolaos Passalis, Anastasios Tefas, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis. Forecasting stock prices from the limit order book using convolutional neural networks. In *2017 IEEE 19th Conference on Business Informatics (CBI)*, pages 7–12. IEEE, July 2017.

[47] Wei Zhang. Shift-invariant pattern recognition neural network and its optical architecture. In *Proceedings of Annual Conference of the Japan Society of Applied Physics*, 1988.

[48] Wei Zhang. Parallel distributed processing model with local space-invariant interconnections and its optical architecture. *Applied Optics*, 29(32):4790–4797, 1990.

[49] Coenraad Mouton, Johannes C. Myburgh, and Marelie H. Davel. Stride and translation invariance in cnns. In Aurona Gerber, editor, *Artificial Intelligence Research*, volume 1342 of *Communications in Computer and Information Science*, pages 267–281. Springer International Publishing, 2020.

[50] Haris Iqbal. Harisiqbal88/plotneuralnet v1.0.0, December 2018.

[51] D. H. Hubel and T. N. Wiesel. Receptive fields and functional architecture of monkey striate cortex. *The Journal of Physiology*, 195(1):215–243, 1968.

[52] Masakazu Matusugu, Katsuhiko Mori, Yusuke Mitari, and Yuji Kaneda. Subject independent facial expression recognition with robust face detection using a convolutional neural network. *Neural Networks*, 16(5):555–559, 2003.

[53] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202, 1980.

[54] K. Fukushima. Neocognitron. *Scholarpedia*, 2(1):1717, 2007.

[55] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[56] Beining Chen, Christoph Mueller, and Peter Willett. Evaluation of a bayesian inference network for ligand-based virtual screening. *Journal of Cheminformatics*, 1(1):5, 2009-04-29.

[57] Ammar Abdo, Naomie Salim, and Ali Ahmed. Implementing relevance feedback in ligand-based virtual screening using bayesian inference network. *SLAS Discovery*, 16(9):1081–1088, 2011-10-01. Publisher: Elsevier.

[58] Ammar Abdo and Naomie Salim. Ligand-based virtual screening using bayesian inference network. In *Library Design, Search Methods, and Applications of Fragment-Based Drug Design*, volume 1076 of *ACS Symposium Series*, pages 57–69. American Chemical Society, 2011-01-01. Section: 4.

[59] Mohammed Khaldoon Altalib and Naomie Salim. Similarity-based virtual screen using enhanced siamese multi-layer perceptron. *Molecules*, 26(21):6669, 2021.

[60] Mohammed Khaldoon Altalib and Naomie Salim. Hybrid-enhanced siamese similarity models in ligand-based virtual screen. *Biomolecules*, 12(11):1719, 2022-11. Number: 11 Publisher: Multidisciplinary Digital Publishing Institute.

[61] Fouaz Berrhail, Hacene Belhadef, and Mohammed Haddad. Deep convolutional neural network to improve the performances of screening process in LBVS. *Expert Systems with Applications*, 203:117–287, 2022-10-01.

[62] Maged Nasser, Naomie Salim, and Hentabli Hamza. Molecular similarity searching based on deep belief networks with different molecular descriptors. In *Proceedings of the 2020 2nd International Conference on Big Data Engineering and Technology*, pages 18–24, 2020.

[63] Maged Nasser, Naomie Salim, Hentabli Hamza, Faisal Saeed, and Idris Rabiu. Improved deep learning based method for molecular similarity searching using stack of deep belief networks. *Molecules*, 26(1):128, 2020.

[64] Yu-Chen Lo, Stefano E Rensi, Wen Torng, and Russ B Altman. Machine learning in chemoinformatics and drug discovery. *Drug Discov. Today*, 23(8):1538–1546, August 2018.

[65] Sereina Riniker and Gregory A. Landrum. Open-source platform to benchmark fingerprints for ligand-based virtual screening. *Journal of Cheminformatics*, 5(1):26, 2013-05-30.

[66] Model the Biosphere | BIOVIA – Dassault Systèmes. `https://www.3ds.com/products-services/biovia/` Accessed : 2023-04-16.

[67] Pipeline pilot - BIOVIA - dassault systèmes®. Accessed in 2023-04-26.

[68] Gym documentation. `https://www.gymlibrary.dev/`. Accessed: 2023-04-28.

[69] Adventure - gym documentation. `https://www.gymlibrary.dev/environments/atari/adventure/`. Accessed: 2023-04-28.

[70] Air raid - gym documentation. `https://www.gymlibrary.dev/environments/atari/air_raid/`. Accessed: 2023-04-28.

[71] Alien - gym documentation. `https://www.gymlibrary.dev/environments/atari/alien/`. Accessed: 2023-04-28.

[72] Amidar - gym documentation. `https://www.gymlibrary.dev/environments/atari/amidar/`. Accessed: 2023-04-28.

[73] Bank heist - gym documentation. `https://www.gymlibrary.dev/environments/atari/bank_heist/`. Accessed: 2023-04-28.

[74] Defender - gym documentation. `https://www.gymlibrary.dev/environments/atari/defender/`. Accessed: 2023-04-28.

[75] Kung fu master - gym documentation. https://www.gymlibrary.dev/environments/atari/kung_fu_master/. Accessed: 2023-04-28.

[76] Bryan Thornbury. Can reinforcement learning be used for classification https://www.brthor.com/blog/ai/reinforcement-learning/can-reinforcement-learning-be-used-for-classification/, 2021-02-02. Accessed: 2023-04-29.

[77] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[78] PPO — stable baselines3 2.0.0a8 documentation. https://stable-baselines3.readthedocs.io/en/master/modules/ppo.html. Accessed: 2023-05-16.

[79] Michael Waskom. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021-04-06.

[80] pandas - python data analysis library. https://pandas.pydata.org/. Accessed: 2023-05-16.

[81] NumPy. https://numpy.org/. Accessed: 2023-05-16.

[82] Ammar Abdo and Naomie Salim. Similarity-based virtual screening with a bayesian inference network. *ChemMedChem: Chemistry Enabling Drug Discovery*, 4(2):210–218, 2009.

[83] Dávid Bajusz, Anita Rácz, and Károly Héberger. Why is tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of Cheminformatics*, 7(1):20, 2015-05-20.