POLITECNICO DI TORINO

Master Degree course in Computer Engineering

Master Degree Thesis

Drug-likeness Prediction and Fragment Extraction using Transformer-based Graph Neural Network on Traditional Chinese Medicine Molecules

Supervisors
Prof. Stefano Di Carlo
Prof. Alessandro Savino
Dr. Roberta Bardini
Ing. Riccardo Smeriglio
Prof. Du Weiwei (Kyoto Institute of Technology)

Candidate
Marco Colangelo

Academic Year 2023-2024

Acknowledgements

I am deeply thankful to my supervisor in Japan, Prof. Du Weiwei, for the incredible opportunity to work as a researcher in Japan, a dream come true.

I would like to express my gratitude to my supervisors in Italy, Prof. Stefano Di Carlo, Dr. Roberta Bardini, and Ing. Riccardo Smeriglio, for their guidance and feedback throughout this project. It has been a long and challenging journey, but the satisfaction I feel now is immense, and your support has been fundamental in helping me achieve this.

To my friends around the world: wherever I went, I found a family to welcome and support me from Potenza, through Torino, and on to Kyoto. I will never forget the moments spent together, the fears, joys, uncertainties, and laughter that I shared with you all. A piece of my heart now lives with each of you across the globe.

To my family, my mother, my father, and my sister: the endless patience in listening to me, supporting me, laughing and crying together despite the thousands of miles and multiple time zones. I would not be who I am today without them and I am eternally grateful for your love and strength. You have been and will always be the supporting pillars of my life.

A special thank you to Akari, the unexpected partner who makes life's journey so much brighter. Your constant presence by my side, even when 10,000 kilometers apart, has been my anchor. You help me preserve my inner child in a world that often rushes to grow up and I cannot thank you enough for that.

Finally, I extend my gratitude to everyone I have met or conversed with along the way, those who have shared so much and even those who have shared so little, whether you have offered advice without knowing me or lent a hand in times of need. Each of you has contributed a fragment, a piece of the puzzle essential to sustain the soul that defines me today and to support the creation of who I will become tomorrow.

Abstract

The use of Traditional Chinese Medicine spans thousands of years, yet its integration into modern pharmaceutical research has been limited [1]. A major challenge is the lack of systematic evaluation of the chemical properties of TCM compounds, which slows their development into approved pharmaceuticals [2]. Adopting drug-likeness as a metric, which refers to the physicochemical and structural properties of a molecule that make it potentially suitable for development as a pharmaceutical drug, is crucial for determining whether a compound could be a viable drug candidate.

Given the diversity and complexity of TCM, manually evaluating each compound for drug-likeness is impractical. Therefore, an efficient, systematic approach is needed to assess the drug-likeness of TCM compounds and understand the chemical structures that contribute to their therapeutic potential. To address this challenge, this thesis proposes a data-driven approach using structured data and machine learning techniques to systematically evaluate the drug-likeness of TCM compounds, enabling the identification of promising candidates for pharmaceutical development.

The strategy involves building a custom Transformer-based Graph Neural Network model to predict drug-likeness by analyzing molecular structures and identifying the most pharmacologically relevant chemical substructures within each compound. ZINC, a curated collection of commercially available chemical compounds specifically designed for virtual screening, is the dataset used for the model's training, validation, and testing. Only compounds from the "in vitro" and "in vivo" categories have been selected. The model achieves an accuracy of 83%, a precision of 80%, and a recall of 88% on the test set.

The ready-to-use model has then been applied to a dataset related to TCM. This enables the model to determine which compounds may be drug-like and offer insights into specific chemical fragments that contribute to drug-likeness, revealing patterns within TCM's unique molecular compositions.

We extracted significant molecular fragments from TCM compounds and identified molecules with promising characteristics. A literature review was conducted to explore the pharmaceutical applications of these molecules, connecting our predictions to known pharmacological data. Out of 147 clusters, 112 have confirmed archetypes or molecules that are closely related to these archetypes, which may be considered as tested or clinically used drugs.

Through this innovative application, the thesis bridges ancient medicinal knowledge and novel computational techniques, opening new possibilities for sustainable drug discovery from natural resources. The extraction of fragments also highlighted the presence of repeated patterns, which could be further examined in future research. The clustering approach enabled us to identify representative compounds with promising drug-like properties, highlighting the potential of integrating TCM compounds into modern pharmaceutical development. This provides a solid foundation for future drug discovery efforts, integrating traditional remedies into modern medicine.

Contents

Acronyms

AD Alzheimer's Disease

ADS Asymmetric Double Sigmoidal

AI Artificial Intelligence

AUC Area Under the Curve

BST Breadth-First Search

CHEBI Chemical Entities of Biological Interest

ChEMBL Chemical European Molecular Biology Laboratory

DBCVI Density-Based Clustering Validation Index

DBSCAN Density-Based Spatial Clustering of Applications with Noise

ECFPs Extended Connectivity Fingerprints

EPA U.S. Environmental Protection Agency

FPR False Positive Rate

GAT Graph Attention Network

GCN Graph Convolutional Network

GNN Graph Neural Network

HDBSCAN Hierarchical Density-Based Spatial Clustering of Applications with Noise

LLM Large Language Model

LSTM Long short-term memory

MeSH National Library of Medicine's Medical Subject Headings

MLP Multilayer Perceptron

MSSGAT Molecular SubStructure Graph ATtention

MST Minimum Spanning Tree

NPASS Natural Product Activity and Species Source

PC Principal Component

PCA Principal Component Analysis

QED Quantitative Estimate of Drug-Likeness

QSAR Quantitative Structure-Activity Relationship

RNN Recurrent Neural Network

ROC Receiver Operating Characteristic

ROC-AUC Area Under the Receiver Operating Characteristic Curve

SMILES Simplified Molecular Input Line Entry System

TCM Traditional Chinese Medicine

TPR True Positive Rate

UMAP Uniform Manifold Approximation and Projection

Chapter 1

Introduction

Traditional Chinese Medicine (TCM) is a well-established and comprehensive healthcare system that has developed over thousands of years, providing an integrative approach to health that balances the body, mind, and environment [1].

TCM takes a different approach compared to Western medicine, which often centers on alleviating symptoms. TCM prioritizes prevention and looks at the body as a whole, aiming for overall wellness [3]. Over centuries, TCM has developed a deep understanding of health, offering various treatments for everything from minor issues to chronic illnesses [3]. It harnesses the power of natural ingredients like herbs, minerals, and even animal products to help the body heal itself and maintain balance [4].

At the heart of TCM is a wealth of botanical resources and intricate herbal formulas [5]. These natural ingredients include various compounds, each with unique pharmacological effects. Together, they serve as the basis for thousands of medicinal recipes found in the TCM pharmacopeia, reflecting a deep understanding of nature's healing potential [5]. Modern research highlights the pharmaceutical potential of TCM, as many drugs derived from natural products play vital roles in medicine today [6]. An exemplary success is the discovery of the anti-malarial drug artemisinin, derived from the TCM herb Qinghao (Herba Artemisiae Annuae) and documented in ancient texts like "A Handbook of Prescriptions for Emergencies" (East Jin Dynasty, around 317-420 A.D.) by Ge Hong [7]. Professor Youyou Tu and his research team, inspired by the ancient medical document, won the Nobel Prize in Medicine in 2015. Their work exemplifies the value of traditional knowledge in addressing contemporary medical challenges [7].

In recognition of its importance, TCM was formally incorporated into the 11th edition of the International Statistical Classification of Diseases in May 2019, establishing it as a globally recognized system of healthcare [7]. This inclusion by the World Health Organization signifies TCM's relevance for modern medical science, underscoring its role in managing both health and disease across the 129 World Health Organization member states worldwide [7].

However, its effectiveness has been the subject of limited studies, creating a scientific gap [8] that this research aims to contribute to filling. Research and development of drugs based on natural products present unique challenges, such as chemical complexity, difficulty of isolation, and variability of natural compounds [9] [10] [11]. Many organic compounds derived from plants are toxic to humans and cannot be used for therapeutic purposes [9]. Therefore, it is critical to be able to predict the pharmacological properties of molecules derived from natural sources to select those with the greatest therapeutic potential and minimize toxicity risks [9].

Beyond historical insights, TCM's potential in pharmacology continues to grow, with databases cataloging the bioactive constituents and metabolites of TCM formulas [12] [13] [14]. Various models and systems have been built to study the TCM leveraging different systems and different technologies, from traditional methods based on the application of classical metrics [15] [16] to more advanced Machine Learning models [17].

This thesis presents a deep learning architecture designed to predict whether a given compound will function as a drug and offer insights into specific chemical fragments that contribute to drug-likeness, revealing patterns within TCM's unique molecular compositions. Due to the lack of labeled data for TCM compounds, the model was initially trained on non-TCM molecule databases and subsequently applied to TCM compounds, ZINC15 [18]. After the training and testing phases, it has been applied to a TCM database, TM-MC 2.0 [13], to find valuable TCM compounds predicted as drug-like. Using data from a TCM database in this way can assist researchers in investigating TCM therapeutic compounds, facilitating modern drug discovery and improving our understanding of complex biological mechanisms.

The structure of this thesis is organized as follows: Chapter 2 provides an overview of the current state of the art, highlighting its limitations and existing approaches to the discussed problem, alongside an introduction to the proposed project solution. Chapter 3 details the construction of the dataset for this project, the architecture adopted for the prediction system, and the post-processing step, showing in detail the mechanism of fragment extraction and the clustering technique. Chapter 4 presents and critically analyzes the model's performance. Finally, Chapter 5 concludes the thesis with a summary of the entire work and a discussion on potential future implementations.

Chapter 2

Background

In this chapter, we provide a comprehensive overview of current methods for analyzing TCM compounds and their applications in pharmaceutical research. We also introduce an innovative approach aimed at overcoming many of the limitations of traditional techniques, highlighting new pathways in TCM research that leverage data-driven and model-based solutions.

2.1 Related Works

The complexity of TCM has led researchers to explore a range of analytical methods. Network pharmacology has been widely used for TCM analysis, integrating pharmacokinetics and systems biology to map out the holistic effects of TCM formulations. Luo et al. (2021) demonstrated its utility in linking TCM components to specific diseases, although capturing the full scope of compound interactions remains challenging [5].

Often used for biological activity prediction, traditional Quantitative Structure-Activity Relationship (QSAR) models struggle with the complexity of multi-component systems like TCM [19]. Zhang et al. (2023) explored hybrid models that combine QSAR with deep learning to better address these limitations [20].

Fingerprints-based models, such as those employing Extended Connectivity Fingerprints (ECFPs)s combined with Multilayer Perceptron (MLP)s, have also been effective for TCM analysis [21]. Despite this, simplifying molecular structures into binary vectors can result in information loss that limits predictive power.

Machine learning models are also increasingly applied to TCM research. Zhu et al. (2021) used supervised learning to predict the activity of TCM compounds, yet the lack of comprehensive, labeled datasets continues to be a major barrier [22]. Wang et al. (2020) highlighted the importance of improving TCM data quality to enhance predictive reliability [23].

Deep learning approaches, especially GNN [24], have shown potential in capturing complex molecular relationships. Jiang et al. (2021) applied GNN to predict drug-likeness in herbal components, demonstrating their superior capacity compared to traditional methods [25]. However, interpretability remains a concern, which has prompted the integration of attention mechanisms like Graph Attention Network (GAT) [26] and Transformer-based GNNs [27] to better highlight crucial features.

Relevant works that inspired this research include the Molecular Substructure Molecular SubStructure Graph ATtention (MSSGAT) [28], which was developed to enhance molecular property identification by focusing on substructures within molecules adopting a multi-modal approach and tree decomposition methodologies. Additionally, educational resources such as those provided by DeepFindr [29] have been instrumental in offering insights into GNNs and their applications in molecular analysis, helping researchers to better understand and address the associated challenges.

2.2 Limitations of Traditional Drug Discovery and Adopting a New Approach

Traditional drug discovery methods, while historically effective, face considerable challenges, especially in sourcing new drugs from complex systems like TCM [8]. These methods typically link each molecule to a specific disease to assess clinical relevance [3]. However, focusing on individual compounds has presented several issues as researchers have begun to address this complexity scientifically:

- **Limited Generalization**: The vast number of diseases makes it impossible to test each compound for every possible condition, especially as new diseases keep emerging. This poses a significant challenge for researchers, as developing a method to identify effective compounds across such a broad and evolving landscape is incredibly difficult [10] [11]. A more efficient system for screening compounds upfront would help narrow down the options, saving time and resources during the testing phase [10] [11].

- **Weak Direct Correlation**: Establishing straightforward links between specific chemical compounds and particular diseases is difficult because TCM compounds often work in a multi-target manner rather than addressing single-disease mechanisms [5]. Many diseases have diverse causes and can manifest differently in patients, further complicating efforts to correlate individual compounds directly with specific conditions [30]. This complexity makes it hard to rely on a simple, uniform approach and instead calls for a more thoughtful method to understand how these compounds impact health.

- **Compound Synergy**: TCM compounds are often prescribed in multi-component formulas where the combined effects are more significant than individual compounds alone, thanks to synergistic interactions [30]. However, current drug discovery and validation methods primarily analyze compounds in isolation, overlooking these synergistic effects [31]. This poses a limitation, as the efficacy of TCM formulations may rely on these combined actions, and isolating compounds does not fully capture their therapeutic potential when used as intended in TCM practices [31] [5]. Focusing on single compounds is still valuable for understanding their individual effects, and with improved methodologies, we aim to balance both targeted and holistic approaches more effectively.

With significant funding and a considerable investment of time, it is possible to gradually uncover the complexities of TCM. Numerous studies continue to add incremental insights [2] [32] [33]. However, a more general and systematic tool is needed to initially screen herb molecules for potential efficacy, aiming to streamline the selection process and limit the number of solutions that require extensive exploration and testing.

This project proposes to use drug-likeness as a general chemical feature, providing an efficient approach for selecting promising molecules and facilitating a more targeted methodology for research in TCM.

2.3 Drug-Likeness: From Static Criteria to Novel Solutions

Drug-likeness refers to the set of chemical and physical characteristics that make a compound likely to be an effective and safe drug [34]. The definition of drug-likeness can vary depending on the biological context being analyzed, with some focusing on properties that ensure good absorption and distribution, while others emphasize factors like target specificity and low toxicity [35]. Generally, a drug-like molecule has properties such as solubility in water and fat, potency at biological target, efficiency of the ligand, and low molecular weight [36]. It involves evaluating whether a compound possesses properties similar to those of known pharmaceuticals, enabling it to interact appropriately with biological targets while maintaining favorable pharmacokinetics and toxicity profiles. Hence drug-likeness generally aims to estimate how 'drug-like' a molecule is, based on its potential to become an efficacious and safe medication, making the drug development process more efficient and increasing the probability of success [35].

2.3.1 Introduction to Common Drug-Likeness Metrics

Traditionally, drug-likeness has been assessed using specific metrics that help predict a compound's potential as a successful drug. These metrics are designed to ensure that a compound has the right balance of chemical and physical properties for biological activity and safety.

Lipinski's rule of five

One of the most well-known sets of criteria is the "Lipinski's rule of five" [37] or "Pfizer's rule of five" or simply "Rule of five", which includes properties like molecular weight, lipophilicity, hydrogen bond donors, and hydrogen bond acceptors. These rules were established to identify compounds with a high probability of becoming orally active drugs. In particular, the rules a molecule must follow to be defined drug-like are:

1. Octanol-water partition coefficient log P ≤ 3
2. Molecular mass < 300 daltons
3. ≤ 3 hydrogen bond donors
4. ≤ 3 hydrogen bond acceptors
5. ≤ 3 rotatable bonds

It provides a clear and effective guideline for identifying compounds with suitable properties for oral bioavailability. This rule, based on empirical data from successful drugs, helps researchers predict whether a molecule is likely to be absorbed well if taken orally.

Quantitative Estimate of Drug-Likeness (QED)

Another method termed QED offers a more flexible evaluation of potential drug candidates compared to traditional rule-based metrics such as Lipinski's Rule of Five. In contrast to these conventional approaches, QED takes into account a broader range of molecular properties and provides a continuous score for drug-likeness, rather than a binary classification. This makes QED a more comprehensive and adaptable instrument for assessing a compound's potential [38].

QED evaluates critical physicochemical properties of molecules to assess their potential as drug candidates. Attributes such as molecular weight, lipophilicity, hydrogen bond donors and acceptors, polar surface area, and the presence of aromatic rings are prioritized. QED employs desirability functions with assigned weights to calculate individual scores for each property. These scores are then combined to generate an overall QED score, which ranges from 0 to 1. Compounds with higher scores exhibit more favorable drug-like characteristics [38]. The QED is calculated using the following formula:

$$QED = \exp\left(\frac{1}{n}\sum_{i=1}^{n} w_i \ln d_i\right) \tag{2.1}$$

Where:

- d_i: Desirability function for the i-th molecular property.
- w_i: Weight assigned to the i-th property.

- n: Total number of molecular properties considered.

Unlike rigid rule-based approaches, QED allows compounds to be ranked by their relative desirability (2.2), making it a valuable tool for prioritizing candidates in drug discovery. The original version is calculated with the formula of an Asymmetric Double Sigmoidal (ADS) Function [38]:

$$d(x) = \frac{1}{1 + \exp(a(x - b))} + \frac{1}{1 + \exp(c(d - x))} \tag{2.2}$$

Where:

- x: The molecular property value.
- a, b, c, d: Parameters that define the shape of the sigmoidal function determined by fitting empirical data from known drug compounds to accurately represent the desirable distribution of molecular properties.

QED also accounts for cases where certain unfavorable properties can be tolerated if other properties are near-optimal, allowing for a more holistic and realistic view of compound quality. Parameters and weights can be calculated based on statistics from previous datasets of chemical compounds or selected according to subjective criteria.

Limitations in the static metrics

The methods discussed became popular over time due to their ease of use and the interpretability of the results they provide.

Overall, while both QED and Lipinski's Rule of Five have their respective benefits, they each have limitations that must be taken into account during drug discovery. QED provides a more flexible and data-driven approach; however, it depends on the quality of the available data [38]. Additionally, there is an element of subjectivity in selecting the empirical data used to compute the parameters within the ADS functions or in the weights assigned to various molecular properties [38]. On the other hand, Lipinski's Rule of Five is easy to use and well-established but can be overly rigid and simplistic, potentially missing out on promising drug candidates, particularly those outside the conventional chemical space [37].

To address these challenges, next-generation metrics driven by data and Artificial Intelligence (AI) are emerging [39] [21] [40] [41]. These advanced approaches leverage vast datasets and machine learning algorithms to predict drug potential with greater accuracy and flexibility. In the following sections, we will explore these data-driven, AI-based metrics and their advantages over traditional models.

2.3.2 Data-Driven and AI-Based Approaches for Drug-Likeness Evaluation

As already introduced before, deep learning models can automatically extract relevant features from data, providing a more comprehensive and insightful analysis of drug-likeness [42].

These models can be implemented using various strategies, such as regression [43] and classification [44]. The regression approach involves predicting the level of drug-likeness as a continuous value, offering a more detailed assessment compared to a binary system [43]. On the other hand, the classification approach categorizes compounds as either "drug" or "non-drug," providing a straightforward method for candidate selection [44].

The input molecular representation for the model can vary:

- **Physicochemical Properties**: Using precomputed properties as input, providing direct information to the model about characteristics relevant to drug-likeness [39].

- **1D Simplified Molecular Input Line Entry System (SMILES) Analysis**: Deep learning models specialized in Natural Language Processing are adapted to conduct analyses on SMILES strings [45].
- **Fingerprints**: Using molecular fingerprints to encode the chemical structure into a vector representation [21].
- **Molecular Graphs**: Representing molecules as graphs, capturing information about atomic connectivity and chemical bonds [46].

The choice of model, molecular representation, and learning strategy depends on the specific application and the nature of the available data. Some popular solutions in computational chemistry are proposed below.

Physicochemical Properties Analysis with Machine Learning approaches

The utilization of comprehensive databases allows researchers to access extensive information concerning the chemical properties of a wide array of molecules. By exploiting this information, traditional machine learning models can be developed to analyze datasets, focusing on the properties of various molecules. These models are capable of predicting drug-likeness by assigning either binary labels or continuous regression values [39].

However, a significant limitation of this approach is its reliance on properties identified and categorized by human expertise. This constraint restricts the model's ability to effectively incorporate the 2D and 3D structural characteristics of the molecules [47] [48]. Consequently, such methodologies may fail to identify novel features that could be crucial for enhancing predictive accuracy and advancing drug discovery efforts. Addressing these limitations is essential for improving the robustness of machine learning applications in the fields of cheminformatics and pharmacology [47] [48].

SMILES for Recurrent Neural Network, Long short-term memory, and Large Language Model

The most common method for representing molecules in a virtual environment is through an encoding system known as the SMILES [40]. This system uses a specific notation to describe the structure of chemical compounds using short ASCII strings. The SMILES notation system, grounded in principles of molecular graph theory, enables precise structure specification through concise and intuitive grammar [40]. Its format is particularly compatible with high-speed computational processing, making it ideal for various chemical applications. This compatibility offers both chemists and computers significant ease of use, facilitating the creation of highly efficient tools, such as unique notation generation, zero-order (constant-speed) database retrieval [19], flexible substructure searches [19] and predictive models for molecular properties [49]. Since SMILES is a text-based encoding, it is natural to utilize technologies like Recurrent Neural Network (RNN) [45], Long short-term memory (LSTM) [45], or Large Language Model (LLM) [50] due to their widespread application in similar tasks. These models are designed to identify patterns within the strings, which correspond to the molecular structures.

However, a significant challenge with this approach is that it often limits our study to the macro-structure of the molecules, translating them into a one-dimensional representation found in text strings. Consequently, the inherent 2D and 3D atomic arrangements typical of molecules are overlooked, even when models attempt to visualize the compounds [51] [52].

Fingerprints and Multilayer Perceptron models

ECFPs are extensively utilized in cheminformatics to characterize molecular structures as fixed-length binary vectors, effectively indicating the presence or absence of specific substructures within a molecule [53]. These fingerprints are created using tools like RDKit [54], which take molecular structures from standard file formats (such as SMILES) and transform them into ECFP vectors. This procedure involves parsing the molecular structure, implementing the ECFP algorithm to derive the fingerprint, and, if necessary, normalizing the data for subsequent machine learning applications [53].

The process of integrating ECFPs with MLP neural networks consists of several key steps [55]: Initially, data preparation requires the generation of ECFP vectors from a dataset of molecules with established properties. Following this, the design of the model architecture defines the number of hidden layers and the count of neurons within each layer in the MLP. During the training phase, ECFP vectors are fed into the MLP, wherein weights are adjusted based on a loss function aimed at enhancing predictive accuracy [55]. Ultimately, validation and testing assess the model's performance on previously unseen data to evaluate its generalization capabilities.

This methodology has been validated in various studies such as CheMixNet [21] [56], which integrates ECFPs with neural networks to forecast chemical properties, demonstrating superior performance compared to traditional techniques.

The combination of ECFPs with MLP models presents a strong framework for predicting molecular properties, with active research aimed at refining these representations [56]. Nonetheless, ECFPs remain fixed vector representations that encapsulate the existence or lack of specific substructures within a molecule; therefore, they may overlook more intricate and nuanced structural details [56] [57]. Hence, there exists a necessity for more dynamic and adaptive approaches that can learn from the entire molecular structures and autonomously extract novel features.

Molecular Graph for Graph Neural Networks

Graph models are specifically designed to process and learn from graph-structured data [24]. Unlike traditional neural networks, which typically work with structured inputs like images (grids of pixels) or sequences (such as text), they are ideal for data represented as graphs, where entities are linked through complex relationships [46] [24].

Graphs are composed of nodes (representing entities) and edges (representing relationships between these entities). This makes the graphs particularly well-suited for problems involving interconnected systems or networks, such as social networks [58], knowledge graphs [59], protein interactions [60] or molecular structures [61]. GNNs are capable of learning both the properties of individual nodes and the relationships within the graph, enabling them to perform tasks like node classification, link prediction, and graph-level classification [62].

In the context of drug discovery, they are especially valuable because molecules can be naturally represented as graphs, where atoms serve as nodes and chemical bonds as edges [25]. This representation allows Molecular Graphs to capture and model the intricate relationships and dependencies between different parts of a molecule, which are critical for determining its chemical and biological properties [25]. By leveraging this graph representation, GNNs can be used to predict molecular properties, identify potential drug candidates, and analyze the interactions between compounds and biological targets.

This thesis has exploited the properties of this technology to extract as much info as possible from the molecules in the TCM database, trying to find a solution for several limitations the GNN models bring with them.

2.3.3 Challenges and Strategies for Effective Use of Graph Neural Networks in Drug Discovery

The use of GNNs in important fields like drug discovery comes with challenges, particularly when it comes to interpretability and sensitivity to training data. GNNs are powerful in highlighting relevant nodes and relationships, but understanding exactly how the model makes its predictions can be difficult, which limits the transparency of the results [63]. Therefore, we decided to adopt a solution equipped with an attention mechanism such as Transformers to improve interpretability.

Another key challenge is that GNNs are sensitive to biases in the training data [64] [22]. If the dataset is unbalanced or incomplete, the model can learn distorted patterns, leading to inaccurate generalizations [65]. Errors in the input data, like incorrect or missing information, can also impact the reliability of the predictions [64]. These challenges highlight the need for careful validation and strategies to mitigate bias, ensuring that GNNs produce reliable outcomes, especially in sensitive applications [65]. To explore and fix this issue, Section 2.4 provides an overview of the research conducted to find the best dataset for training so that we can build the most effective model possible.

2.4 Dataset Selection: A Critical Overview

Finding a suitable dataset for training machine learning models is particularly challenging in fields like drug discovery and molecular property prediction, especially when dealing with TCM compounds [66]. The quality and nature of the available data directly influence the model's accuracy, generalizability, and reliability, which makes the process of dataset selection and evaluation a critical factor in achieving robust predictive capabilities [66].

However, discovering appropriate datasets for TCM leads to unique difficulties. TCM compounds are characterized by their complexity and multi-component nature, which is not always well represented in conventional databases. The majority of the existing databases present limitations such as insufficient data on pharmacological efficacy, the absence of standardized encoding formats like SMILES, or poor data quality. These challenges make it difficult to effectively seek TCM compounds for model training and testing.

In this section, we explore the challenges faced in identifying and evaluating the datasets suitable for our project. We describe the difficulties encountered in sourcing relevant data for model training, validation, and testing, along with the obstacles in ensuring the quality and consistency of information.

2.4.1 Data Quality Concerns

The search for an appropriate dataset is a comprehensive process, involving multiple factors to ensure that the data accurately represents the distinction between drug-like and non-drug-like molecules. It addresses several issues that may emerge in various data sources:

- **Lack of Information on Pharmacological Efficacy**: Many databases, although focused on TCM, did not provide direct information on the pharmacological efficacy of the molecules. For instance, TC-MC 2.0 [13] offered a wide range of data but did not directly link compounds to biological activity, making it difficult to evaluate the actual therapeutic impact of these molecules.

- **Absence of SMILES Encoding**: SMILES [40] encoding is crucial for representing and processing molecular structures. However, some databases, such as Dr. Duke's Phytochemical and Ethnobotanical databases [67], did not include SMILES encoding, limiting the usability of their data for training machine learning models.

- **Difficulty in Accessing Data**: Some databases presented access challenges or were available in impractical formats. For example, the LOTUS data database [68], although comprehensive, was only available in MongoDB or SDF formats, which required specialized software for processing and extracting the information we needed. Another example is TCMBank [69]: even if full of useful information about TCM compounds, the website lacked a downloadable dataset.

- **Unclear or Inaccurate Data**: In some cases, the data was unclear or inconsistent. For example, TCMSID [70] included a "Drug-likeness" metric, but without a clear definition, making it difficult to interpret. Additionally, the classification of compounds as "drug" or "non-drug" in datasets such as TCMBank [69] was unreliable, necessitating further validation from other sources.

- **Lack of Negative Datasets**: Even in databases that reported pharmacological activity [71] [72] [73] [70], there was often a lack of examples explicitly recognized as devoid of biological activity, making it challenging to create a balanced dataset for machine learning model training.

All the information acquired during the research was processed and reported in the tables 2.1 and 2.2.

Database Name	**TCM Scope**	**DB Download**	**Chemical Info**	**2D Structure**	**SMILES**	**PubChem Link**
TCMSP db [72]	Yes	No	Yes	Yes	Yes	Yes
LOTUS data [68]	No	Yes	Yes	Yes	Yes	Yes
TM-MC 2.0 db [13]	Yes	Yes	Yes	Yes	Yes	Yes
IUPHAR/BPS Guide to Pharmacology [73]	No	Yes	Yes	Yes	Yes	No
ChEBI [74]	No	No	Yes	Yes	Yes	No
Dr. Duke's Phytochemical and Ethnobotanical databases [67]	Yes	No	No	No	No	No
ChEMBL [75]	No	Yes	Yes	Yes	Yes	Yes
The Natural Products Atlas [76]	No	Yes	Yes	No	Yes	No
TCMSID [70]	Yes	No	Yes	No	Yes	Yes
EPA DSSTox database [77]	No	Yes	Yes	Yes	Yes	Yes
PubChem BioAssays [78]	No	Yes	Yes	Yes	Yes	Yes
ZINC15 [18]	No	Yes	Yes	Yes	Yes	Yes
DrugBank [79]	No	Yes	Yes	Yes	Yes	Yes

Table 2.1: Overview of TCM databases and their basic features.
Column Descriptions: *Database Name* - Name and reference of the database; *TCM Scope* - Indicates if the database includes TCM compounds (Yes/No); *DB Download* - Availability of downloading the database (Yes/No); *Chemical Info* - Availability of detailed chemical information (Yes/No); *2D Structure* - Availability of two-dimensional molecular structures (Yes/No); *SMILES* - Availability of SMILES notation (Yes/No); *PubChem Link* - Availability of a direct link to the corresponding PubChem entry (Yes/No).

Database Name	**Drug-likeness metric**	**Natural Compounds?**	**Total Compounds Available**	**Of which natural**	**Drug-Disease Link**	**Note**
TCMSP [72] db	No	Yes	12144	12144	Yes	Indicates whether the compound is associated with a disease in the context of Traditional Chinese Medicine.
LOTUS data [68]	No	Yes	276518	276518	Not present	
TM-MC 2.0 db [13]	Yes, score between 0 and 1	Yes	34349	34349	Not present	
IUPHAR/BPS Guide to Pharmacology [73]	Yes, but it is not clear for all the databases	Not only	12592	419	Yes	Indicates whether the compound is an approved or investigational drug and the diseases for which it is used or studied.
ChEBI [74]	No	Not only	32667	2194	Not present	
Dr. Duke's Phytochemical and Ethnobotanical databases [67]	No	Yes	29585	29585	Yes	Indicates plants used in traditional medicine to treat various diseases.
ChEMBL [75]	Yes, more like a filter than a real metric	Yes	2.4M	32667	Yes	Indicates bioactive molecules and their therapeutic targets, including links to diseases. Several valuable sections
The Natural Products Atlas [76]	No	Yes	Not specified	Not specified	Not present	Not so interesting, more focused on bacteria and fungi.
TCMSID [70]	Yes, binary variable, but it is not clear what it means.	Yes	20015	20015	Not present	
EPA DSSTox database [77]	No, but a list of usages and bioactivities	Not only	11000	Not specified	More like a list of usages and bioactivities	More for toxic compounds.
PubChem BioAssays [78]	Not specified	Yes	Not specified	Not specified	Not present	Quite generic, assays are several lists of compounds organized according to their objective.
ZINC15 [18]	Yes	Yes	>750M	224205	Yes	ZINC is divided into several subsets, including the "in-vivo" and "in-vitro" subsets that are of interest to the project
DrugBank [79]	Yes	Yes	> 500k	Not specified	Yes	

Table 2.2: Overview of TCM databases and their detailed features.
Column Descriptions: *Database Name* - Name of the database along with its reference citation; *Drug-likeness metric* - Indicates whether the database provides a metric to assess the drug-likeness of compounds and additional details if applicable; *Natural Compounds?* - Specifies if the database includes natural compounds; *Total Compounds Available* - The total number of compounds available in the database; *Of which natural* - The number of natural compounds within the total compounds available; *Drug-Disease Link* - Indicates whether the database provides links between drugs and diseases; *Note* - Additional relevant information or comments about the database.

2.5 Graph Neural Network Methodologies

In this section, we explore different GNN methodologies popular for molecular analysis. Various GNN technologies have been developed, each with distinct characteristics that make them suitable for specific tasks [80] [26] [27] [81]. These technologies differ in their approaches to message passing [82], convolutional layers [80] and attention mechanisms [26] [27], allowing for diverse modeling capabilities depending on the complexity of data such as the chemical compounds. Below, we present different types of GNNs and their methodologies in the context of drug discovery and molecular property prediction.

2.5.1 Graph Neural Network (GNN)

GNNs [24] are a class of machine learning models specifically designed to work with graph-structured data. Unlike traditional neural networks, which work with structured data like grids of pixels in images or sequential text, GNNs are well-suited for problems involving complex relationships and interactions, such as those found in social networks [58], molecular structures [61], and knowledge graphs [59]. The fundamental advantage of GNNs lies in their ability to effectively capture and process the relational information among graph nodes through iterative message passing and aggregation mechanisms [46].

A graph, denoted as $G = (V, E)$, is composed of nodes V and edges E that represent the entities and their interactions, respectively. In GNNs, each node i has an initial feature vector $h_i^{(0)}$, which is iteratively updated to represent the node's contextual information based on its neighbors. The core idea behind GNNs is to propagate and aggregate information from neighboring nodes, which ultimately allows each node to gather an enriched representation of its local structure.

The message-passing mechanism, which is the key operation in GNNs, can be mathematically formulated as follows:

$$h_i^{(k+1)} = \sigma\left(W^{(k)} h_i^{(k)} + \sum_{j \in \mathcal{N}(i)} f(h_i^{(k)}, h_j^{(k)}, e_{ij})\right) \tag{2.3}$$

Here, $h_i^{(k)}$ represents the feature vector of node i at layer k, $W^{(k)}$ is a learnable weight matrix at layer k, and $\mathcal{N}(i)$ represents the set of neighboring nodes of i. The function f aggregates the information from neighboring nodes, and e_{ij} denotes the features of the edge connecting nodes i and j. The activation function σ (such as ReLU) is applied element-wise to introduce non-linearity into the model.

In our model, atoms are represented as nodes and bonds as edges, allowing the GNN to handle complex interactions within molecules. The flexibility and expressiveness of GNNs make them a powerful tool for capturing relationships and dependencies in graph data, ultimately enabling effective feature learning for downstream predictive tasks [24].

2.5.2 Graph Convolutional Network (GCN)

A Graph Convolutional Network (GCN) [80] is a type of GNN that utilizes convolutional operations to extract information from the molecular graph, capturing the relationships between atoms and their properties.

The GCN update rule can be expressed as:

$$H^{(k+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(k)} W^{(k)}\right) \tag{2.4}$$

where $\tilde{A} = A + I$ is the adjacency matrix with added self-loops, $\tilde{D}$ is the diagonal degree matrix of $\tilde{A}$, $H^{(k)}$ is the node feature matrix at layer k, $W^{(k)}$ is a learnable weight matrix and σ

is the activation function. This normalization with $\tilde{D}$ helps to maintain stability during training by controlling the magnitude of aggregated information.

The adjacency matrix A is defined such that the element A_{ij} equals 1 if nodes i and j are connected and 0 otherwise. Additionally, $H^{(0)}$ typically contains the initial features of the nodes, while each subsequent layer refines these features by aggregating information from neighboring nodes (example of application in Figure 2.1).

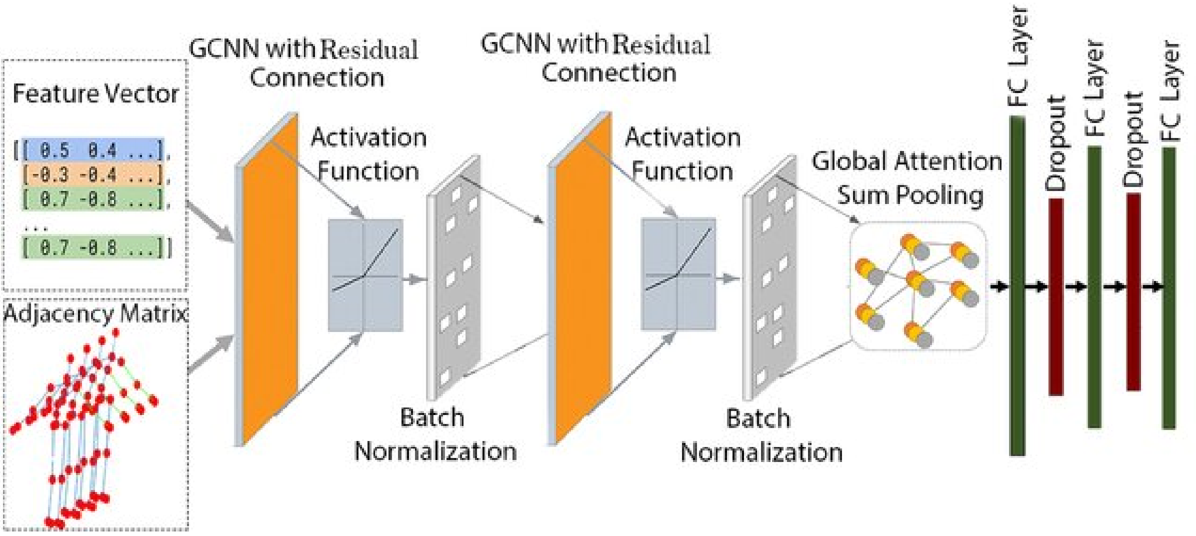


Figure 2.1: Concept and Image of GCN architecture with residual connections and global attention pooling taken from Bari et al, 2021, [80]. The model processes node features and adjacency matrix through stacked GCN layers with batch normalization, residuals, and an attention-based pooling layer, followed by fully connected and dropout layers for final prediction.

2.5.3 Graph Attention Networks (GAT)

GAT [26] are built upon Graph Convolutional Networks (GCNs) by introducing an attention mechanism. This mechanism evaluates the importance of each node during message passing, enabling the model to prioritize specific connections and more effectively capture relevant information.

The attention coefficient between two nodes i and j is computed as:

$$e_{ij} = a\left([Wh_i \parallel Wh_j]\right) \tag{2.5}$$

where h_i and h_j are the hidden representations of nodes i and j, $\parallel$ represents concatenation, W is a learnable weight matrix, and a is a single-layer feedforward neural network followed by a LeakyReLU activation function.

The attention coefficients are then normalized across all neighboring nodes using the softmax function:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}(i)} \exp(e_{ik})} \tag{2.6}$$

where $\mathcal{N}(i)$ represents the neighbors of node i.

The final output for each node is computed by aggregating the features of its neighbors, weighted by the attention coefficients:

$$h_i' = \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} W h_j\right) \tag{2.7}$$

where σ denotes a non-linear activation function, such as ReLU (example of implementation in Figure 2.2).

This approach enables GATs to assign different weights to neighboring nodes, enhancing the model's ability to capture complex structures in graph data.

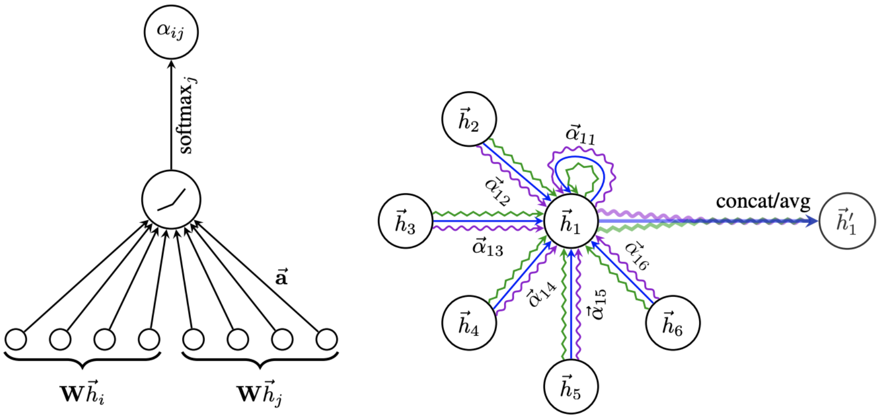


Figure 2.2: Concept and Image of GAT architecture taken from Veličković et al., 2018, [26]. The left diagram demonstrates the attention mechanism, where attention coefficients α_{ij} are calculated using softmax over neighboring nodes' embeddings. The right diagram shows how each node $\vec{h}_1$ aggregates information from its neighbors $\vec{h}_2, \vec{h}_3, \ldots, \vec{h}_6$ weighted by the learned coefficients α_{ij}, followed by concatenation or averaging to produce the updated node embedding $\vec{h}_1'$. The label "concat/avg" refers to the operation on the heads in case a MultiHead approach is adopted.

2.5.4 Transformer-Based Graph Convolutional Networks

Transformer-based GCNs [27] combine the principles of GCNs with the attention mechanism of Transformers, allowing for more efficient information passing between nodes by capturing the global importance of connections.

Transformers are a neural network architecture designed for processing sequential data by leveraging an attention mechanism [83]. Initially developed for natural language processing tasks [83], Transformers have since become a versatile tool across many areas of deep learning [84]. The main innovation of Transformers is their ability to assess the importance of different parts of the input sequence, allowing them to concentrate on the most relevant information irrespective of its position [83] [84]. Unlike traditional recurrent networks, which analyze sequences step-by-step, Transformers can process all elements of a sequence simultaneously, which makes them highly efficient for parallel computing. Their effectiveness largely stems from the use of "self-attention", which allows the model to capture relationships between different parts of a sequence, resulting in robust feature extraction and representation learning [84] [83] (see Figure 2.4a). Figure 2.3 presents an example of implementation, taken from the paper [28].

The self-attention mechanism is defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.8}$$

where Q, K, and V are the query, key, and value matrices derived from the node features, and d_k is the dimensionality of the key vectors.

The query, key, and value matrices are computed as:

$$Q = H^{(k)}W_Q, \quad K = H^{(k)}W_K, \quad V = H^{(k)}W_V \tag{2.9}$$

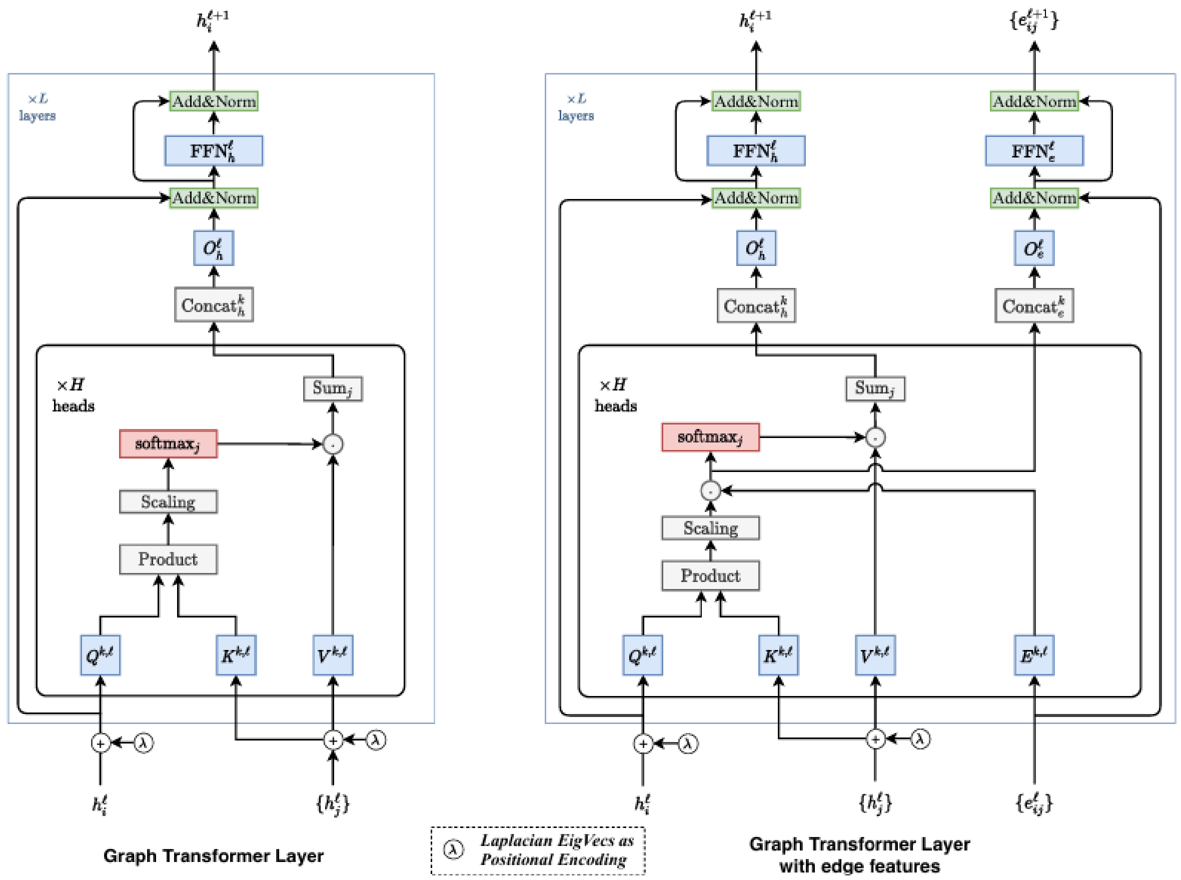


Figure 2.3: Concept and Image of Transformer-GCN architecture taken from Dwivedi et al., 2021, [27]. Both Graph Transformer layers share core components, including multi-head attention mechanisms to capture complex node relationships, Add and Norm layers to stabilize and regularize learning, and Feed-Forward Networks (FFN) to refine node embeddings. Each layer integrates positional encoding, enabling spatial awareness within the graph. The left diagram focuses on node features alone, while the right extends the model with edge features to improve relational learning between nodes.

where W_Q, W_K, and W_V are learnable weight matrices. The self-attention output is then used to update the node features, which allows the model to capture global dependencies and relationships within the graph.

To capture the diverse aspects of relationships in the data, Transformer models utilize multi-head attention. This process involves executing multiple attention mechanisms simultaneously, each using its own set of weight matrices. The outputs from these parallel attention heads are then concatenated and linearly transformed to generate the final output.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h)W_O \tag{2.10}$$

where each $\text{head}_i = \text{Attention}(QW_{Q_i}, KW_{K_i}, VW_{V_i})$, and W_O is a learnable weight matrix (Figure 2.4b).

The final output of the Transformer layer is computed by applying a feed-forward network to the self-attention output:

$$H^{(k+1)} = \text{FFN}(\text{Attention}(Q, K, V)) \tag{2.11}$$

where FFN represents a feed-forward neural network, typically consisting of two linear transformations with a ReLU activation in between.

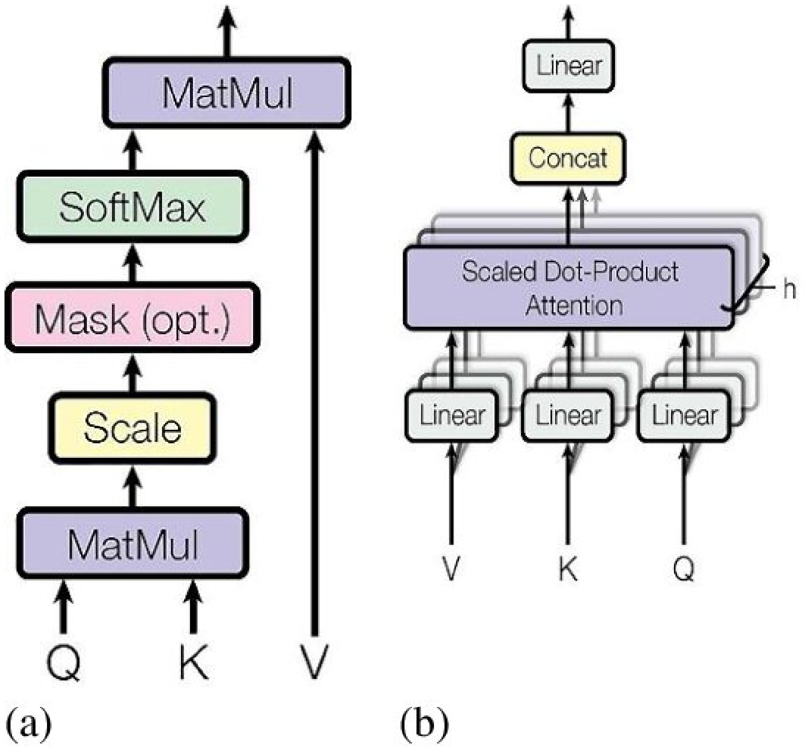


Figure 2.4: (a) Self-attention mechanism defined in 2.8. (b) Multi-head attention mechanism defined in 2.10. (Images taken from Vaswani et al, 2017, [83]).

2.6 Dimensionality Reduction

Dimensionality reduction is a crucial technique in data analysis and machine learning, aimed at simplifying complex datasets by reducing the number of variables under consideration. This process not only helps mitigate the curse of dimensionality but also enhances computational efficiency, facilitates data visualization, and can improve the performance of predictive models by eliminating noise and redundant features. By transforming high-dimensional data into a lower-dimensional space, dimensionality reduction techniques enable the extraction of meaningful patterns and insights that might be obscured in the original high-dimensional space.

Various methods have been developed for dimensionality reduction, each with its unique strengths and suitable applications. These methods can be broadly categorized into linear and non-linear techniques. Linear methods, such as PCA, assume that the underlying structure of the data can be captured through linear combinations of the original variables. In contrast, non-linear methods, like UMAP, are designed to capture more complex, non-linear relationships within the data. In this section, we focus on PCA and UMAP.

2.6.1 PCA

PCA is a dimensionality reduction method designed to simplify complex datasets while retaining as much essential information as possible. It achieves this by generating new variables, known as principal components, which are uncorrelated and ordered to capture the maximum variance in the data. The process involves solving an eigenvalue/eigenvector problem to determine these principal components, which are defined based on the specific dataset rather than predetermined, making PCA a highly adaptable technique for data analysis [85].

Mathematical Formulation To perform PCA, the data matrix X (composed of n observations and p variables) is first centered by subtracting the mean of each feature, resulting in a zero-mean dataset X_{centered}:

$$X_{\text{centered}} = X - \mathbf{1}_n \mu^T \tag{2.12}$$

In this expression, $\mathbf{1}_n$ is an n-dimensional column vector of ones, and μ is a p-dimensional column vector containing the mean of each variable.

The covariance matrix C of the centered data is then computed as:

$$C = \frac{1}{n-1} X_{\text{centered}}^T X_{\text{centered}} \tag{2.13}$$

The next step is to identify the principal components by solving the eigenvalue problem:

$$Cv_i = \lambda_i v_i \tag{2.14}$$

Here, v_i are the eigenvectors representing the directions of maximum variance in the data, while λ_i are the eigenvalues corresponding to the amount of variance explained by each eigenvector.

Finally, the data can be transformed into a lower-dimensional space by projecting it onto the principal components. This is achieved through the following operation:

$$Z = X_{\text{centered}} W \tag{2.15}$$

where W is the matrix formed by the eigenvectors associated with the largest eigenvalues, and Z is the dataset represented in the reduced-dimensional space.

PCA is a powerful tool for simplifying datasets, allowing for better visualization and interpretation while minimizing the loss of important information. Selecting the principal components that capture the most variance, effectively reduces the complexity of the data while maintaining its underlying structure. However, PCA has several limitations. It assumes linear relationships between variables and is sensitive to scaling and outliers, which can distort results. PCA also requires a sufficiently large sample size for stable results and cannot handle missing data without imputation, which may introduce biases [85]. These limitations should be considered when applying PCA, as they can affect the validity and interpretability of the results.

2.6.2 UMAP

To represent data structures, UMAP constructs a weighted graph where each data point connects to its nearest neighbors, with weights assigned based on pairwise distances. A high-dimensional fuzzy topological structure is created, and UMAP then seeks a low-dimensional embedding that preserves this structure by minimizing cross-entropy between the high- and low-dimensional representations. It is a non-linear dimensionality reduction technique that maps high-dimensional data into a lower-dimensional space while preserving both local and global structures. UMAP is based on principles from topology and manifold learning, assuming that data lies on a Riemannian manifold [86] and is organized through local connectivity [87].

Mathematical Formulation In a high-dimensional space, the probability of connecting two points i and j is defined as:

$$\mu_{i|j} = \exp\left(-\frac{d(i,j) - \rho_i}{\sigma_i}\right)$$

where $d(i,j)$ represents the distance between points i and j, ρ_i is the distance from point i to its nearest neighbor, and σ_i is a normalization factor that adjusts for local density. The symmetric probability between points i and j is given by:

$$\mu_{ij} = \mu_{i|j} + \mu_{j|i} - \mu_{i|j} \cdot \mu_{j|i}$$

In the corresponding low-dimensional space, UMAP models the connection probability as:

$$\nu_{ij} = \left(1 + a \cdot d_{ij}^{b}\right)^{-1}$$

where d_{ij} is the distance between points i and j in the reduced space, and a and b are hyperparameters controlling the shape of the model. To align the structures of the high- and low-dimensional spaces, UMAP minimizes the following cross-entropy objective:

$$C = \sum_{i \neq j} \left[\mu_{ij} \log\left(\frac{\mu_{ij}}{\nu_{ij}}\right) + (1 - \mu_{ij}) \log\left(\frac{1 - \mu_{ij}}{1 - \nu_{ij}}\right) \right]$$

This optimization ensures that the low-dimensional embedding accurately preserves the manifold structure of the data, capturing intricate relationships even in sparse, high-dimensional datasets.

2.7 Clustering Technique

Clustering is a fundamental technique used to group similar data points based on their inherent characteristics. By partitioning data into distinct clusters, clustering algorithms help identify patterns, structures, and relationships within complex datasets without requiring prior knowledge of the groupings.

In our research, we employ clustering techniques to analyze drug-like molecules identified by our model. We therefore present KMeans and HDBSCAN as clustering methods.

2.7.1 KMeans

The k-means clustering algorithm partitions a dataset of n observations into k clusters, where k is a hyperparameter. Each observation is assigned to the cluster with the nearest centroid, aiming to reduce intra-cluster variance. Through an iterative process, the algorithm updates the cluster centroids and reassigns observations to optimize this objective [88].

Mathematical Formulation Given a set of observations $\{x_1, x_2, \ldots, x_n\}$, where each observation is a d-dimensional vector, k-means clustering seeks to partition the data into k subsets $S = \{S_1, S_2, \ldots, S_k\}$ by minimizing the within-cluster sum of squared distances (WCSS):

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2$$

Here, μ_i is the centroid of cluster S_i, calculated as:

$$\mu_i = \frac{1}{|S_i|} \sum_{x \in S_i} x$$

where $\| \cdot \|$ represents the Euclidean norm, and $|S_i|$ denotes the number of points in cluster S_i. The objective of this method is to minimize the total squared distance between points and their respective cluster centroids, thereby reducing the overall within-cluster variance.

2.7.2 HDBSCAN

HDBSCAN extends traditional density-based clustering methods by introducing a hierarchical approach that overcomes the limitations of single-density threshold methods like Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [89]. Instead of producing a flat clustering, HDBSCAN generates a hierarchy of clusters, allowing for the identification of structures at multiple density levels [90].

The algorithm starts by calculating the core distance for each data point, defined as the minimum distance required to include a specified number of neighbors (MinPts). Using these core distances, a mutual reachability distance is defined between pairs of points, which balances density-based relationships and spatial distances. This information is represented as a Minimum Spanning Tree (MST) of the mutual reachability graph.

HDBSCAN simplifies this hierarchy by focusing on significant clusters. This is achieved by a stability-based measure that evaluates the persistence of clusters across different density levels. Stability is defined in terms of the relative excess of mass, capturing how long a cluster remains intact as the density threshold changes. The algorithm then extracts a set of optimal clusters by maximizing the overall stability of the resulting structure.

This chapter established a foundation for understanding the current landscape and emerging trends in TCM compound analysis. By integrating advanced AI-driven methodologies and addressing data-related challenges, we set the stage for the subsequent branch in Chapter 3, where we will detail how these innovative strategies performed as we researched them.

Chapter 3

Materials and Methods

In this chapter, we outline the materials and methodologies used throughout this research project, detailing the datasets, computational tools, and experimental protocols applied. The primary aim of this study was to develop a robust model capable of distinguishing between drug-like and non-drug-like compounds, with a particular focus on TCM compounds. To achieve this, advanced machine learning methods, including deep learning architectures and graph-based techniques, were applied to analyze the molecular structures and predict drug-likeness.

The methods section includes the preprocessing steps for preparing the data to train the machine learning model, the feature extraction approaches used to represent the molecules, and the training procedures for the machine learning model. Each stage of the project, from data acquisition to model evaluation, is described in detail to ensure transparency and reproducibility of the results.

By providing a comprehensive overview of the tools and methodologies used, this chapter aims to clarify how we addressed the challenges associated with analyzing complex chemical compounds and to offer insight into how these methods contributed to the project's objectives.

3.1 Datasets

To ensure a comprehensive and balanced analysis, two distinct datasets were utilized: ZINC15 [18], a large, publicly available database of chemical compounds, and TC-MC 2.0 [13], a dataset specifically built for TCM compounds. These datasets were selected to facilitate the training and evaluation of the model in different contexts. ZINC15 served as the primary training dataset for assessing general drug-likeness, while TC-MC 2.0 was used to test the model's applicability to TCM compounds after the model had been trained and validated.

In the following subsections, we introduce the characteristics and relevance of each dataset, highlighting their content, sources, and how they were used in the model development process.

3.1.1 ZINC15

ZINC [18] is a publicly accessible, open-source database that provides a comprehensive collection of commercially available chemical compounds, primarily designed for virtual screening applications. The database focuses on readily purchasable compounds, making it an invaluable resource for researchers interested in acquiring and experimentally testing specific molecules. ZINC 15, the latest version of the database, contains over 230 million compounds in 3D formats ready for molecular docking, along with more than 750 million compounds available for analog discovery [18].

ZINC has demonstrated substantial utility in drug discovery, providing researchers with detailed access to a vast collection of commercially available compounds and offering relevant data for virtual screening and guide optimization [91] [92] [93]. This makes it a key tool in facilitating the early stages of the drug discovery pipeline.

Further providing 3D structures, ZINC offers a group of additional data relevant to research, including SMILES encoding, computed chemical properties, and information on biological activity, derived from well-established databases such as ChEMBL [75] and DrugBank [79]. The inclusion of biological activity and biogenicity data extends the utility of ZINC, enabling researchers to prioritize molecules with potential therapeutic value.

The ZINC interface is designed to be accessible to individuals with limited knowledge of cheminformatics. It offers several search options, including similarity and substructure searches, which are based on metrics like Tanimoto similarity [94]. Moreover, users can apply various filters to refine their research according to specific criteria such as availability, chemical properties, and biological activity. These features improve the database's usefulness for drug discovery.

One of ZINC's key features is its organization into subsets that group compounds with similar characteristics [18]. This categorization allows researchers to efficiently select compounds most suitable for their research needs, tailoring the selection process to the context of their specific objectives. Specifically, the database includes "in vivo" and "in vitro" subsets, which distinguish between compounds tested in living organisms and those studied in isolated systems. We have exploited these two subsets in particular to train, validate, and test our model. A peculiar characteristic of these two subsets is the large number of available samples, and the ability to differentiate between molecules that can be tested on living organisms and those that cannot, which represents a significant acceleration in the timeline of real-life testing. We used the "in-vivo" subset to represent the positive examples for training and the "in-vitro" subset as the pool of negative examples. This approach allowed us to effectively differentiate between molecules with the potential to be tested in living organisms and those with limited applicability, providing a balanced and relevant training dataset for the model. In this way, we aimed to teach the model to develop its understanding of drug-likeness by learning to differentiate between compounds with high therapeutic potential and those less suitable for testing in living organisms.

In prior discussions, we addressed the challenges of identifying an ideal database for our project, considering issues such as the lack of pharmacological efficacy information, absence of SMILES encoding, or poor data quality in some existing databases. Although ZINC does not specifically target compounds used in TCM, it could still serve as a valuable resource for evaluating drug-likeness in general, particularly through the use of the "in-vivo" and "in-vitro" subsets. Hence, we adopted it to train the model and then applied it to the TCM dataset TM-MC2.0 [13].

3.1.2 TM-MC2.0

TM-MC2.0 is a specialized database designed to provide detailed information on chemical compounds used in TCM as documented in the Korean, Chinese, and Japanese pharmacopeias [13]. In addition to chemical compounds, TM-MC2.0 includes valuable data on prescriptions, gene targets, modern diseases, and their associations, making it an important resource for exploring the therapeutic potential of TCM.

The dataset contains information on 34,349 compounds, providing comprehensive details for each compound, including 2D SDF structure, SMILES encoding, PubChem ID when available, and various chemical properties, including a Drug-likeness score. TM-MC2.0 is particularly advantageous for research focused on TCM due to its specific emphasis on traditional herbal medicine compounds. It also offers detailed chemical data, such as structural information and SMILES encoding, which are essential for cheminformatics and machine learning applications. The availability of a downloadable version of the dataset further enhances its usability by allowing offline analysis [13].

Despite its strengths, TM-MC2.0 also has limitations. One significant limitation is the lack of a direct link between individual compounds and specific diseases, which complicates the study of compound-disease relationships. Furthermore, while the dataset includes a Drug-likeness score for each compound, the exact definition of this metric is not explicitly provided. The score appears to be derived from the QED method, which assigns a value between 0 and 1 to indicate the degree of drug-likeness, with a higher score signifying a stronger similarity to known drugs [38]. It is worth noting that the QED-based method used in TM-MC2.0 may not be entirely suitable for evaluating natural compounds, particularly those with high sugar content, as the weights were trained on a dataset of known drugs. This limitation could affect the accuracy of assessing the drug-likeness of TCM-derived compounds. Despite these issues, TM-MC2.0 served as a valuable validation set for testing our model, which was initially trained on the broader ZINC15 dataset.

After establishing a foundational understanding of drug-likeness through training on ZINC15, the model was subsequently tested on TM-MC2.0 to evaluate its applicability to TCM compounds. This two-step approach ensured that the model could generalize from a broad dataset to a specialized context, effectively discerning drug-likeness within a diverse set of natural compounds.

3.2 Building the Database

The preprocessing of the datasets was a critical step to ensure compatibility with the machine learning models and the reliability of the outcomes. The primary dataset used in this research was ZINC15, which required several preprocessing steps to extract meaningful features, clean erroneous data, and format it appropriately for model training and evaluation. On the other hand, TM-MC2.0 was used as a freely accessible database, first to test the final ready-to-use model in a black-box manner, validating its predictions on TCM compounds, and subsequently as a starting point for a bibliographic review to confirm the predictions. In the confirmation phase, we also used resources like PubChem [78], ChEBI [74], ChEMBL [75], DrugBank [79], and various academic papers.

The first step was to acquire the necessary data. We obtained ZINC IDs from the ZINC15 database, which provided access to a wide range of commercially available chemical compounds. Using these ZINC IDs, we retrieved the corresponding SMILES representations directly from the ZINC15 database. SMILES encoding is essential for representing chemical structures in a machine-readable format, making it suitable for processing by machine learning models.

Labeling the dataset was a vital step in distinguishing between drug-like and non-drug-like compounds. The “in vivo” subset from ZINC15 was labeled as positive (‘Drug = 1‘), representing compounds with potential therapeutic effects, while the “in vitro” subset was labeled as negative (‘Drug = 0‘). These subsets were then merged into a single labeled dataset, facilitating a meticulous classification task for the machine learning model.

To ensure the integrity of the “in vivo” and “in vitro” classes, overlapping samples were removed from the “in vitro” subset. This step was performed to prevent data leakage, which could lead to artificially inflated model performance metrics. Removing shared samples ensured that the positive and negative classes remained distinct, providing a robust foundation for training a reliable model. A molecule could be present in both subsets because data from in-vitro experiments are recorded first, and once these experiments are completed, the compound may be promoted to "in vivo" testing. For this reason, we removed overlapping molecules only from the "in vitro" subset.

A balanced dataset is essential to prevent model bias towards one class. The merged dataset of “in vivo” and “in vitro” compounds was imbalanced, with significantly more negative samples than positive ones. To address this, we undersampled the larger class (in vitro) to match the number of samples in the smaller class (in vivo). This balanced dataset ensured that the model received equal representation from both classes, reducing the risk of skewed predictions. In this way, we obtained a total of 103500 samples per class.

The dataset was then divided into training, validation, and test sets to support model development. Initially, 80% of the data was allocated for training, while the remaining 20% was reserved for validation and testing. This 20% portion was further divided equally into validation and test sets, with each receiving 10% of the total data. This splitting process ensured that the model could be trained on one part of the data, validated on another to tune hyperparameters, and finally tested on a separate, unseen dataset to evaluate its generalization capabilities. The splitting was carefully performed to maintain a balanced representation of both positive and negative classes across the different sets. This process involved random undersampling from the more frequent class. All the steps previously cited are reported in Figure 3.1.

The data acquisition and preparation steps outlined here were essential to prepare the ZINC15 dataset for subsequent stages of model training and evaluation, ensuring that the data was clean, well-labeled, balanced, and properly formatted for effective learning. The TM-MC2.0 dataset remained unaltered, as it was directly downloadable via an API from the official website, complete with SMILES representations of the molecules. It was exclusively used to validate the final trained model, without any preprocessing modifications.

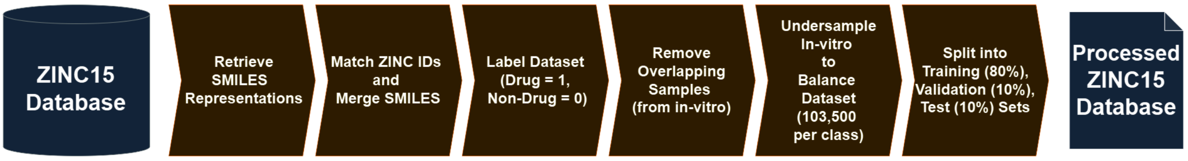

ZINC15 Database
Retrieve SMILES Representations
Match ZINC IDs and Merge SMILES
Label Dataset (Drug = 1, Non-Drug = 0)
Remove Overlapping Samples (from in-vitro)
Undersample In-vitro to Balance Dataset (103,500 per class)
Split into Training (80%), Validation (10%), Test (10%) Sets
Processed ZINC15 Database

Figure 3.1: A step-by-step illustration of the preprocessing workflow for the ZINC15 dataset. The process includes retrieving SMILES representations, matching and merging ZINC IDs, labeling the dataset for drug-like (1) and non-drug-like (0) compounds, removing overlapping samples (from in-vitro), balancing the dataset through undersampling (103,500 per class), and finally splitting the dataset into training (80%), validation (10%), and test (10%) sets to form a new ZINC15 dataset for model development.

3.3 Feature Extraction

In this section, we describe the feature extraction process used to represent the molecules of interest. At this stage, we focused on extracting information related to the nodes and edges of molecules, obtaining a detailed graphical representation of the chemical structures. In graph theory, nodes represent individual entities, while edges represent the connections or relationships between these entities. In the context of molecular graphs, nodes represent atoms, and edges represent chemical bonds connecting these atoms. This featurization process is crucial for providing accurate and complete data to machine learning models, thereby improving their ability to predict the biological and pharmacological properties of the molecules.

Effective feature extraction is crucial for capturing both local atomic interactions and broader molecular structures, which are essential for accurate property prediction. To achieve this, we utilized a combination of traditional node and edge feature extraction methods alongside advanced techniques like tree decomposition.

Tree decomposition, in particular, plays a significant role in simplifying complex molecular graphs by breaking them down into hierarchical structures. This transformation facilitates the analysis of subgroups and the relationships between different atoms, enabling the model to better capture functional groups and molecular interactions. By integrating both raw graph-based features and tree-structured representations, our approach leverages comprehensive insights into molecular properties, enhancing the model's predictive capabilities.

The subsequent subsections detail the specific methodologies used for node and edge feature extraction, followed by an in-depth exploration of the tree decomposition process.

3.3.1 Node and Edge Feature Extraction

The extraction of features from the nodes and edges of molecules was carried out using RDKit [54], a widely used library for manipulating chemical structures and DeepChem [95], a library for chemical features investigation. RDKit allowed us to convert SMILES strings into detailed molecular structures, providing an initial representation of the molecules on which to base feature extraction. We extracted key information such as the number of atoms, bond types, formal charge, and overall molecular topology using DeepChem. See Table 3.1 and 3.2 for details.

We extracted various features to represent each molecule comprehensively, focusing on atom-level and bond-level characteristics that are crucial for predicting molecular properties. At the atomic level, features such as atom type, formal charge, and the presence of lone pairs were captured to provide detailed information about individual atoms. For bonds, features like bond type (e.g., single, double, aromatic) were extracted to describe the connectivity between atoms. These extracted features were then encoded into vectors following conventions set by the library used, making them suitable for input into the model architecture for further processing. This feature extraction and encoding process enhances the model's ability to learn complex chemical relationships effectively.

Each node in the graph represents an atom, described by a feature vector (atom type, formal charge, explicit hydrogens, etc.). The edges represent the bonds between atoms and include information such as bond type, bond order, and stereochemistry.

An important part of the process was managing cases where the molecular structure was invalid according to RDKit, could not be processed, or was made up of only one atom. In that case, the molecule in question was simply discarded from the dataset.

Finally, the feature extraction process was designed to be scalable and efficient through parallelization. We used the concurrent.futures [96] module to parallelize the featurization of a large number of molecules, significantly reducing the time required to process large datasets and improving the efficiency of the pre-processing pipeline.

Table 3.1: Node feature extraction: atom-level features and their descriptions.

Feature	**Description**
Atom type	A one-hot vector of this atom, “C”, “N”, “O”, “F”, “P”, “S”, “Cl”, “Br”, “I”, “other atoms”.
Formal charge	Integer electronic charge.
Hybridization	A one-hot vector of “sp”, “sp2”, “sp3”.
Hydrogen bonding	A one-hot vector of whether this atom is a hydrogen bond donor or acceptor.
Aromatic	A one-hot vector of whether the atom belongs to an aromatic ring.
Degree	A one-hot vector of the degree (0-5) of this atom.
Number of Hydrogens	A one-hot vector of the number of hydrogens (0-4) that this atom connected.

3.3.2 Tree Decomposition

Tree decomposition, also called junction tree decomposition, is a crucial step in representing molecular structures in a way that facilitates efficient analysis of their subgroups and the relationships between different atoms. By transforming the original molecular graphs into hierarchical structures, tree decomposition captures relationships between atom cliques, enabling a clearer understanding of functional groups, molecular interactions, and dependencies. We implemented a Tree Decomposition algorithm to significantly simplify complex molecular structures

Table 3.2: Bond-level features used in the edge feature extraction process

Feature	**Description**
Bond type	A one-hot vector of the bond type: “single”, “double”, “triple”, or “aromatic”.
Same ring	A one-hot vector of whether the atoms in the pair are in the same ring.
Conjugated	A one-hot vector of whether this bond is conjugated or not.
Stereo	A one-hot vector of the stereo configuration of a bond.

and potentially enhance the ability of machine learning models to predict properties accurately. The concept and algorithm we implemented took inspiration from the previous work of Ye et al. [28]. You can find a flowchart that better explains the pipeline in Figure 3.3 and a visual example in Figure 3.2.

Overview of the Tree Decomposition Process

The Tree Decomposition process was conducted using a junction tree decomposition method [28], which helps simplify the representation of complex molecules by identifying groups of atoms that are highly interconnected (referred to as cliques). A MST is then constructed to connect these cliques, ensuring that the overall structure is simplified while retaining key chemical information. This approach reduces the complexity of highly interconnected molecular graphs while preserving important relationships and structural motifs.

The decomposition was implemented by extracting features for both tree nodes and tree edges:

- **Tree Nodes**: Each clique of atoms, formed by identifying rings and highly connected non-ring atom groups, is treated as a node in the tree structure. Shared atoms between cliques ensure proper continuity and connectivity in the graph. The intentional repetition of shared atoms across adjacent cliques ensures proper connectivity between nodes, a requirement of the tree decomposition process. In this way, the decomposition satisfies the running intersection property, ensuring that shared atoms between cliques are included in all relevant cliques, preserving structural continuity and connectivity within the tree structure. Then, the atom-level features within each clique are aggregated, creating a representative feature vector that captures the average properties of its atoms. This allows for a comprehensive representation of each molecular substructure, summarizing its essential characteristics.

- **Tree Edges**: The edges between cliques, which represent bonds linking different molecular substructures, are also processed. Bond-level information between cliques is aggregated, summing the contributions to retain crucial details such as bond type, order, and stereochemistry. This ensures that the nature of the connectivity between different cliques is preserved, providing an accurate representation of molecular connectivity.

Minimum Spanning Tree

In our project, we utilized MSTs to represent molecular graphs in a simplified yet informative way. A MST of a weighted graph is a subset of edges that connects all vertices, forming a spanning tree with the lowest possible total weight. This property makes MSTs ideal for reducing the complexity of molecular graphs while preserving their essential connectivity. By applying Kruskal's algorithm [97] through the scipy library [98], we efficiently extracted the minimum spanning tree for each molecular graph. This approach allowed us to focus on the most critical structural connections while eliminating redundant information.

The resulting graphs were stored using the DGLMolTree library [99], which preserves the atomic and bond information in a hierarchical tree structure suitable for further analysis.

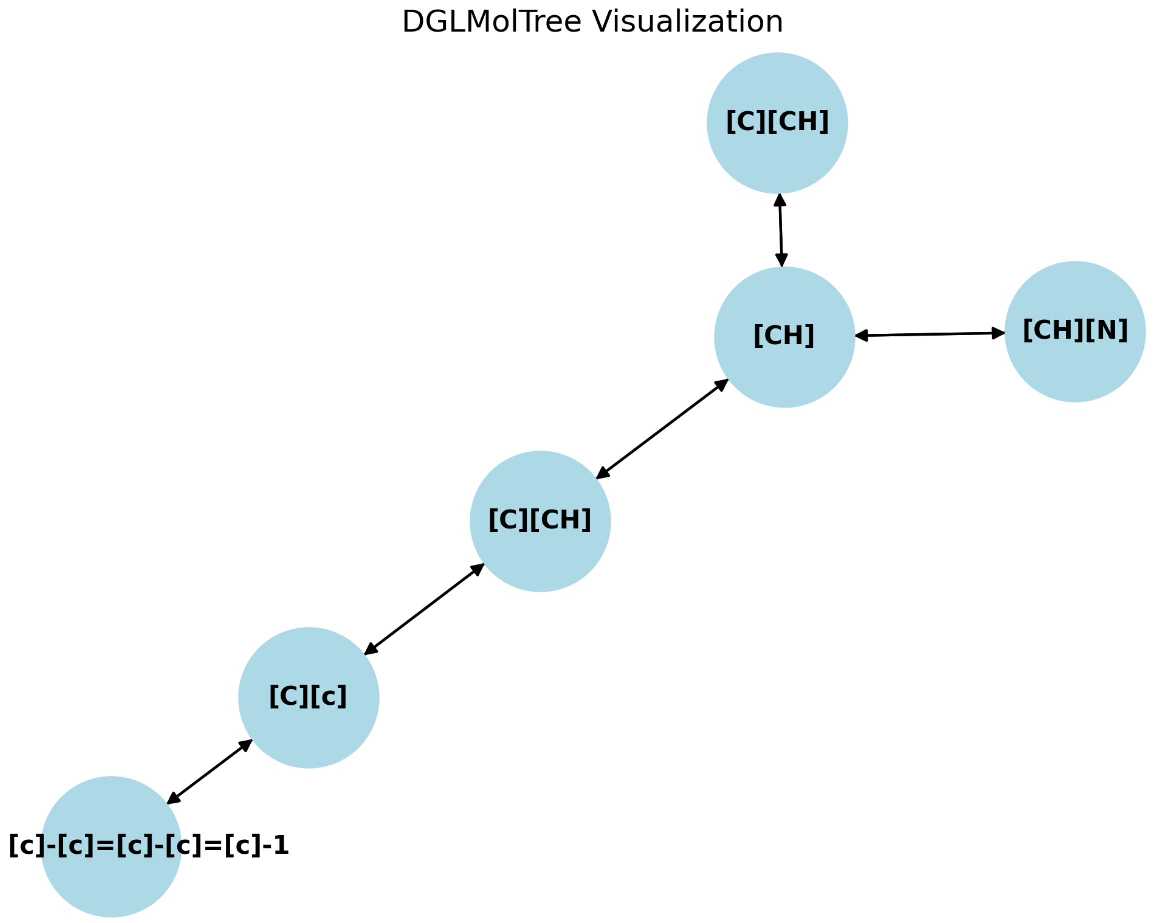


Figure 3.2: Example of a MST generated using the tree decomposition algorithm applied to the molecule `C[C@@H](N)Cc1ccccc1`. In the SMILES string, uppercase `C` denotes non-aromatic carbon atoms, lowercase `c` indicates aromatic carbon atoms, `N` represents a nitrogen atom, and `[C@@H]` specifies a chiral carbon center with defined stereochemistry. Each node represents a molecular fragment (clique), and the edges indicate relationships between them, forming a MST that maps the molecular connections. The repetition of atoms (e.g., `CH`) across adjacent cliques ensures topological continuity and satisfies the running intersection property, preserving the molecular structure during the graph construction.

Error Handling and Robustness

Our tree decomposition algorithm incorporates various strategies to ensure robustness and handle errors effectively when dealing with complex molecular structures. During molecule conversion, it checks for parsing success and discards invalid molecules. For clique extraction, the algorithm includes fallback mechanisms to manage cases where substructures cannot be fully sanitized, allowing it to bypass sanitization if needed to preserve essential molecular connectivity. This approach helps maintain the integrity of the decomposition process while accommodating special cases.

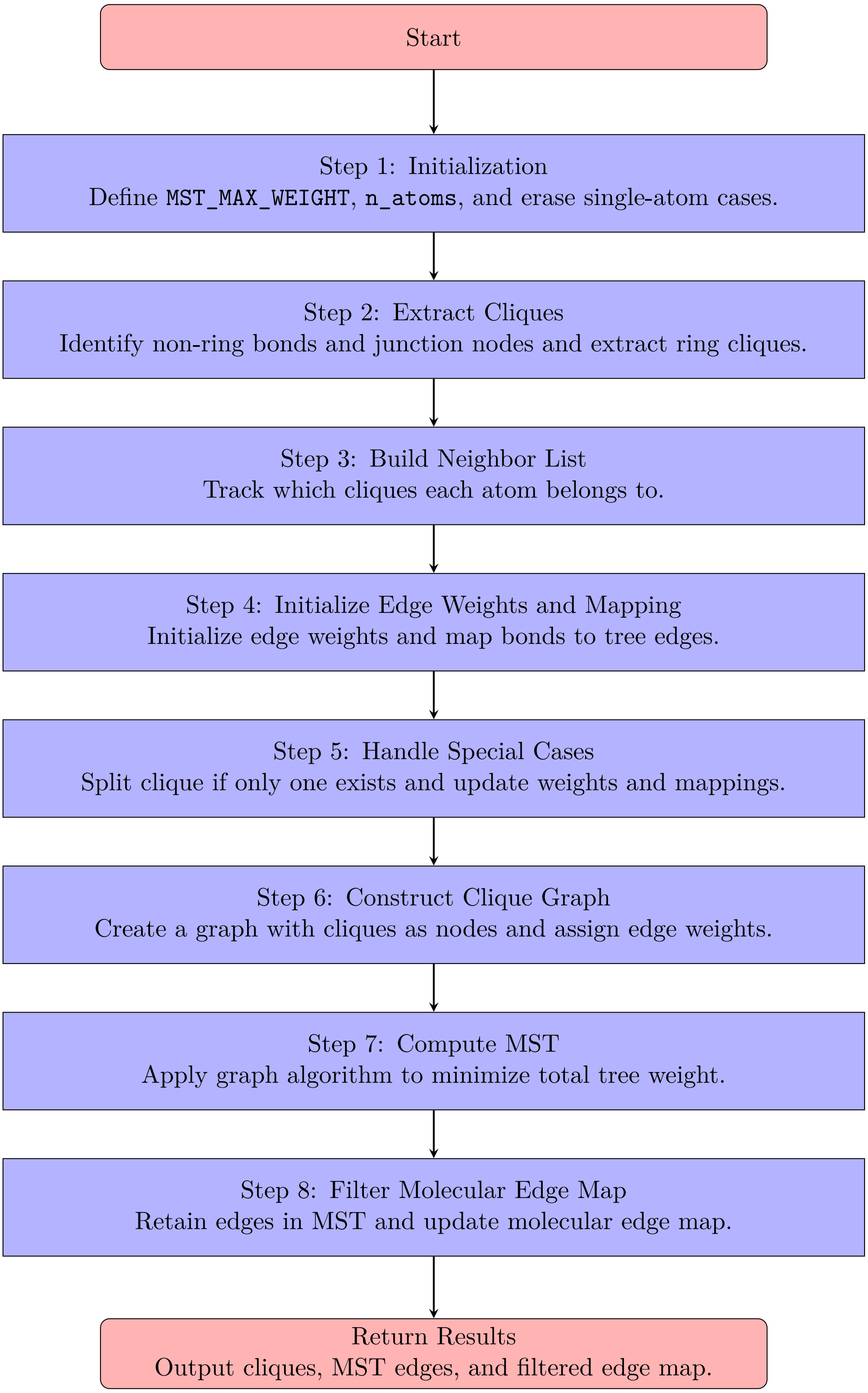

Start
Step 1: Initialization
Define MST_MAX_WEIGHT, n_atoms, and erase single-atom cases.
Step 2: Extract Cliques
Identify non-ring bonds and junction nodes and extract ring cliques.
Step 3: Build Neighbor List
Track which cliques each atom belongs to.
Step 4: Initialize Edge Weights and Mapping
Initialize edge weights and map bonds to tree edges.
Step 5: Handle Special Cases
Split clique if only one exists and update weights and mappings.
Step 6: Construct Clique Graph
Create a graph with cliques as nodes and assign edge weights.
Step 7: Compute MST
Apply graph algorithm to minimize total tree weight.
Step 8: Filter Molecular Edge Map
Retain edges in MST and update molecular edge map.
Return Results
Output cliques, MST edges, and filtered edge map.

Figure 3.3: Tree Decomposition Pipeline for Molecular Graphs: This flowchart outlines the systematic steps involved in decomposing a molecular graph into a tree structure.

Special handling is implemented for minimal structures, such as molecules with only one atom or a single clique. For single-atom molecules, the algorithm returns a single-node tree immediately, while single cliques are split to ensure a valid tree structure. If isolated atoms or disconnected subgraphs are detected, the `DGLMolTree` package adds additional edges to maintain connectivity.

Edge mapping between the tree structure and the molecular graph is validated to ensure all bonds are accurately represented. If a bond lacks specific information in the `mol_edge_map`, the algorithm assigns default attributes to retain the edge. During MST construction, high-edge weights are used as fallbacks to maintain critical clique connections, ensuring a coherent tree structure. These measures enable the algorithm to handle diverse molecular graphs reliably, supporting consistent performance even with unconventional structures.

3.4 Model Architecture

This section presents the architecture of our model, which is designed for molecular property prediction. Integrating advanced GNN methodologies, the model captures both local atomic interactions and broader molecular structures. The architecture processes raw molecular graphs and tree-structured representations, using dedicated GNN encoders to effectively model complex molecular connectivity.

Key GNN techniques, including GCN, GAT, and Transformer-based GCN, were considered for this study. Each offers distinct strengths in processing molecular graph data, as detailed in Section 2.5. After considering what methodology to use, we selected the Transformer-based GCN for its advantages in capturing complex, long-range dependencies [27]. The following subsections outline the model's components, from feature extraction and aggregation to attention mechanisms and pooling, which together create a comprehensive framework for accurate molecular analysis.

We selected a Transformer-based GCN for this study due to its ability to effectively capture long-range dependencies within molecular graphs, a critical feature when analyzing interactions between distant atoms or functional groups [27]. The self-attention mechanism of the Transformer architecture enables dynamic weighting of node importance, making it easily explainable and well-suited for the complex topologies of molecular structures.

This model's ability to handle diverse and intricate molecular features enhances its adaptability to complex molecular topologies, making it particularly well-suited for analyzing large datasets in drug discovery, where understanding the contribution of specific molecular substructures is essential.

3.4.1 Overall Structure

The implemented model architecture is a comprehensive GNN framework designed for molecular data analysis, specifically targeting tasks such as molecular property prediction. This architecture integrates multiple advanced components, including transformer-based convolutional layers, attention mechanisms, pooling strategies, and MLPs to effectively capture and process the intricate relationships within molecular structures. The design emphasizes modularity, scalability, and adaptability, making it well-suited for handling complex molecular datasets. Below is an in-depth breakdown of the model's architecture.

The architecture is forked into two primary pathways:

- **Raw Molecular Graphs Processing (GATencoder_raw)**: Handles the processing of raw molecular graph data, capturing fundamental atomic and bond-level interactions.
- **Tree-Structured Molecular Data Processing (GATencoder)**: Manages the processing of tree-structured representations of molecules, capturing higher-order structural

relationships that might not be evident in raw graphs.

The outputs from these two pathways are subsequently concatenated, transformed, normalized, and fed into an MLP-based classifier to generate the final predictions. This dual-pathway approach ensures that the architecture leverages both atomic-level and structural-level data to provide a comprehensive analysis of molecular properties. By integrating raw graph and hierarchical tree information, the model gains a robust understanding of both local and global molecular features. Graphical representation of the entire architecture in Figure 3.7.

3.4.2 GNN Encoders

GNN Encoder for Raw Molecular Graphs (GATencoder_raw)

GATencoder_raw has been designed to process raw molecular graph data, capturing essential atomic and bond interactions within the molecule. This is achieved using our custom **GNN** Encoder, which incorporates transformer-based convolutional layers from the PyTorch Geometric library [100]. The architecture is visually represented in Figure 3.4. These layers enhance the model's capacity to learn complex patterns by attending to multiple regions of the graph simultaneously.

The adopted components are listed below:

- **Transformer-based convolutional layers**: Incorporate multi-head attention mechanisms, enabling the model to focus on different parts of the graph simultaneously. This allows the capture of complex dependencies between nodes and their neighbors, highlighting subtle interactions that standard convolutional layers might miss.
- **Batch normalization**: Normalizes inputs after linear transformations to stabilize and accelerate training. This helps ensure faster convergence and mitigates issues related to internal covariate shifts.
- **Rectified Linear Unit (ReLU) activation function**: Introduce non-linearity into feature representations through transformations, enhancing the model's capability to learn complex mappings and relationships.
- **Pooling layers**: Reduce graph size by retaining the most significant nodes based on learned importance scores. This hierarchical feature extraction step efficiently captures both global and local information, improving generalization.

GNN Encoder for Tree-Structured Molecular Data (GATencoder)

GATencoder processes tree-structured representations of molecules to capture higher-order structural relationships that may not be evident in raw graphs. This is essential for understanding complex molecular architectures where interactions between different parts of the molecule can influence its overall properties.

Utilizes the same **GNN** class of **GATencoder_raw**, ensuring a consistent approach to processing different molecular representations while accommodating tree-specific structural details. The difference lies exclusively in the type of features provided as input to the layer.

3.4.3 Feature Aggregation and Transformation

After processing raw and tree-structured molecular data through their respective GNN encoders, the model performs the following operations to integrate and transform the extracted features:

- **Feature Concatenation**: The outputs from both GNN encoders (x_t from **GATencoder** and x_r from **GATencoder_raw**) are concatenated along the feature dimension.

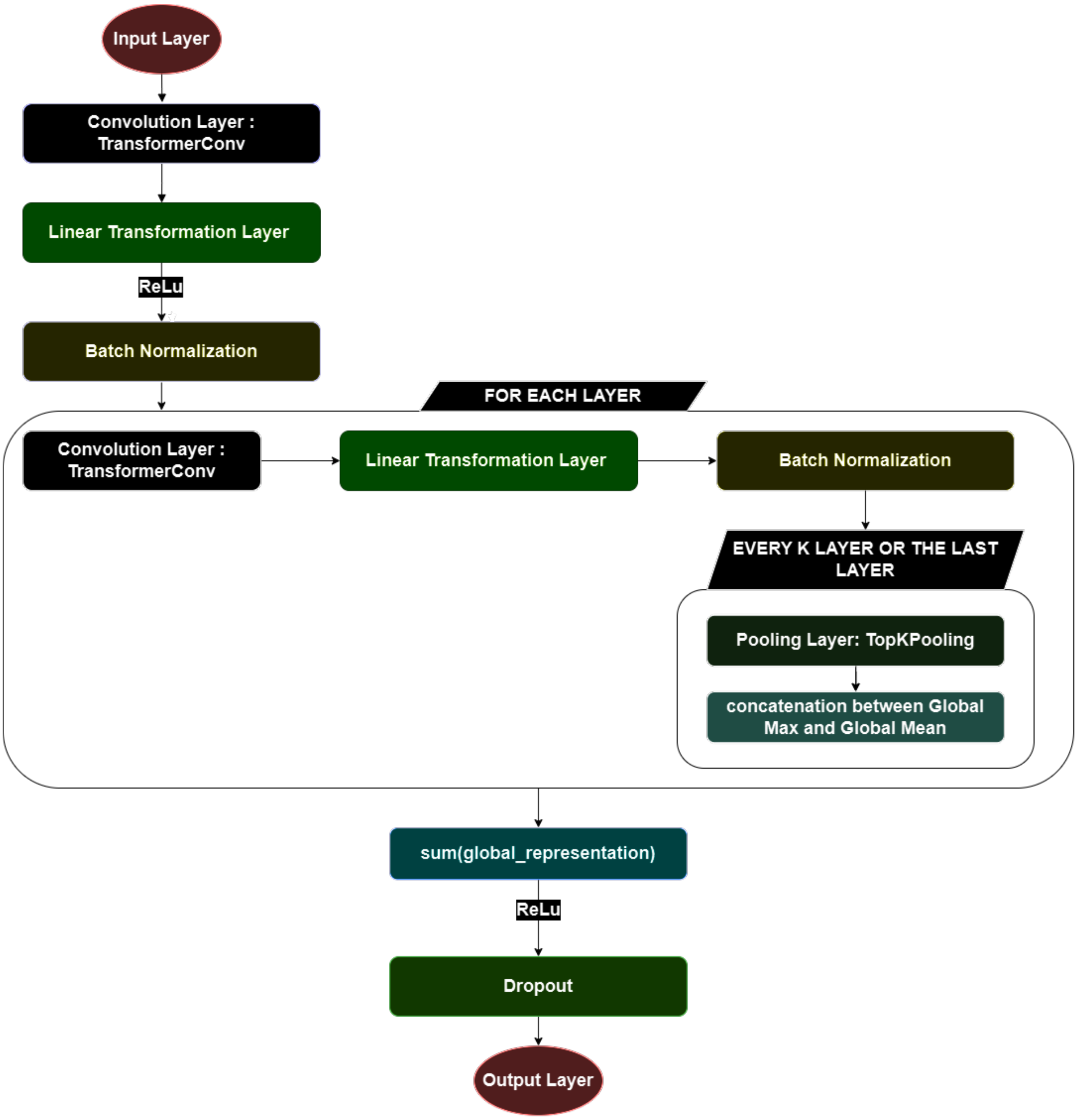

Input Layer
Convolution Layer : TransformerConv
Linear Transformation Layer
ReLu
Batch Normalization
FOR EACH LAYER
Convolution Layer : TransformerConv
Linear Transformation Layer
Batch Normalization
EVERY K LAYER OR THE LAST LAYER
Pooling Layer: TopKPooling
concatenation between Global Max and Global Mean
sum(global_representation)
ReLu
Dropout
Output Layer

Figure 3.4: Architecture representation of the custom GNN Encoder implemented in the model. The structure incorporates Transformer-based convolutional layers, linear transformation layers, batch normalization, and pooling mechanisms to capture and process complex molecular features. Key operations include multi-head attention, hierarchical feature extraction, and concatenation of global max and mean pooling to retain crucial information at each layer. This setup is based on a modified version of the model described in [29].

This results in a combined feature vector that encapsulates information from both molecular representations, allowing the subsequent layers to leverage comprehensive insights.

- **Linear Transformation**: To manage the increased dimensionality resulting from concatenation, the combined feature vector undergoes a linear transformation. This reduces the feature size to a more manageable dimension, ensuring that subsequent processing layers can operate efficiently.

- **Layer Normalization and ReLU Activation**: The transformed features are normalized using layer normalization and passed through a ReLU activation function to introduce non-linearity. This step is crucial for maintaining stable training and enhancing the model's learning capabilities [101] [102].

3.4.4 Classifier Module

The Classifier Module operates as the final classification layer, predicting molecular properties based on the aggregated features. We built it as a multi-layer perceptron with hidden layers and dropout regularization. The total number of layers equals the sum of the number of uploaded features and the hyperparameter for hidden layers.

The components included are enumerated below:

- **Linear Layer**: Receives the transformed feature vector, typically having dimensions that are a multiple of the embedding size. It then applies a linear transformation using a weight matrix and bias term, producing an output vector for further processing or predictions.

- **Hidden Layers**: Contains one or more hidden layers, each followed by a ReLU activation and dropout. The dropout mechanism helps to prevent overfitting by randomly deactivating a subset of neurons during training.

- **Output Layer**: Provides the final prediction, expressed as a binary classification, i.e. either 1 (drug-like) or 0 (not drug-like). Unlike the other components, which are repeated multiple times as specified by the algorithm, this component is positioned at the end of the pipeline and is exploited only once per molecule to finally transform the compound's encoding.

3.4.5 Pooling Strategies

Pooling layers play a significant role in reducing the complexity of graph data while preserving essential information. The architecture employs two primary pooling strategies:

Top-K Pooling (TopKPooling): This strategy reduces the graph size by retaining only the top-k important nodes based on the attention scores learned during training. This hierarchical approach allows the model to capture multi-scale features and focus on the most critical parts of the molecular structure.

Global Pooling (global_max_pool and global_mean_pool): These pooling methods aggregate node features after the graph has been processed. Global max pooling captures the most significant features by taking the maximum value across nodes, while global mean pooling ensures that the overall distribution of node features is represented. This dual pooling strategy ensures that the model learns both prominent and aggregate features, providing a comprehensive view of the graph.

The globally pooled features from each pooling layer are summed or concatenated to form a comprehensive representation that integrates information from multiple pooling stages. This combined representation captures diverse aspects of the molecular structure, supporting robust predictions.

3.4.6 Attention Mechanism

The attention mechanism in our framework serves as a key tool for interpreting and visualizing the significance of different molecular bonds within complex chemical structures. By using attention scores derived from a graph-based machine learning model, such as a Transformer, we can effectively identify the most influential fragments of a molecule. Since these attention scores fall within the range of 0 to 1, we have decided to consider fragments with a score higher than 0.80 as influential. This process provides valuable insights into the model's decision-making process and enhances our understanding of molecular properties and behaviors. For more technical details regarding the attention mechanism in Transformer-Based GNN, refer to the related description in the Subsection 2.5.4.

Mapping Attention Scores to Molecular Bonds

The next step is to utilize the attention mechanism to visually highlight the most significant molecular fragments identified as drug-like by the model. This helps in understanding the justification behind the model's predictions and provides insights into the chemical features that are key drivers of biological activity. The process of highlighting these fragments involves several key steps:

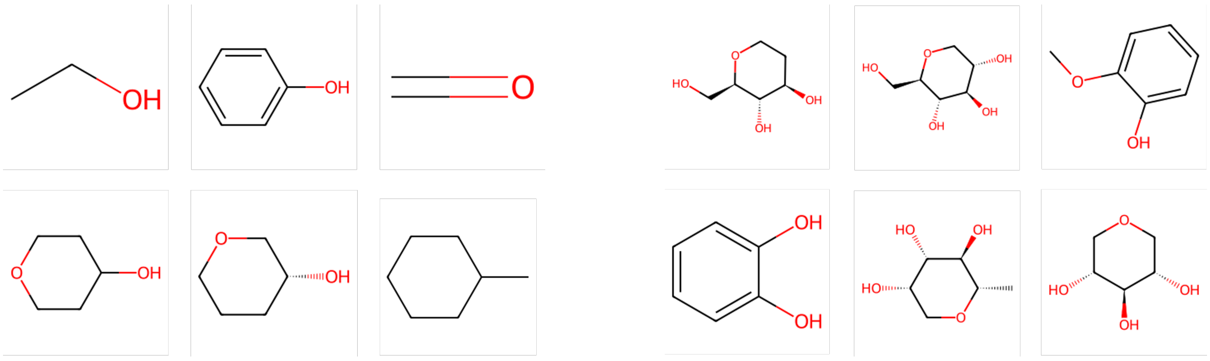
1. **Tree Decomposition of the Molecular Graph:** To manage the complexity of large molecules, we first retrieve the associated tree structures, computed during the pre-processing phase executing the previously cited Tree Decomposition algorithm on each molecule in input (see Subsection 3.3.2).

2. **Association of Attention Scores with Tree Edges:** The attention scores from the model, specifically from the transformer block of the **GATencoder**, which utilizes tree structures as input, are primarily linked to the edges of the tree structure rather than being directly associated with individual bonds. Each tree edge connects two cliques, and the corresponding attention score, normalized to a standardized range (e.g., 0 to 1), reflects the significance of the interaction between these cliques in the context of the model's predictions.

3. **Mapping Tree Edge Scores to Molecular Bonds:** To translate the attention scores back to the original molecular structure, we create a mapping between tree edges and the molecular bonds they represent, along with the corresponding connected groups of atoms. This is achieved by identifying the specific bonds within each clique that correspond to the connections indicated by the tree edges. As a result, each molecular bond inherits an attention score based on its association with the tree edge scores.

4. **(Optional) Deep Search Modality for Clique Aggregation:** The **deep_search** modality was implemented and used during the prediction phase to extract fragments from the molecules. This mechanism employs a Breadth-First Search (BST) traversal algorithm to systematically identify and aggregate closely connected high-attention bonds, effectively merging them into larger, cohesive fragments. By leveraging the hierarchical structure of the BST, the **deep_search** modality ensures that bonds are aggregated based on their connectivity and spatial proximity within the molecular graph (examples presented in Figure 3.5). However, we decided not to proceed with this modality during the post-processing phase. Despite this, it remains a useful feature for those wishing to explore its application in improving interpretability.

5. **High-Attention Fragments Extraction**: The model identified fragments relevant to the final prediction (eventually aggregated in bigger portions through the **deep_search** functionality), which we consider chemically important based on the Drug-Likeness criteria



(a) Examples of fragments extracted with the deep_search feature disabled

(b) Examples of fragments extracted with the deep_search feature enabled

Figure 3.5: A comparison of the fragments extracted using different methods of highlighting. Notice how elaborated the extracted fragments are when the deep_search modality is enabled.

learned during training. Now these fragments are extracted and saved in a dedicated file, ready for eventual further analysis.

6. **Visualization of Highlighted Molecules:** Utilizing RDKit's molecular drawing capabilities, we render the molecule with bonds highlighted according to colors. Colors are assigned based on the attention scores associated with the bonds: white for bonds with scores below 0.50, yellow to orange for scores between 0.50 and 0.80, and red for scores above 0.80. This visualization allows for an intuitive and immediate understanding of which parts of the molecule are considered most significant by the attention mechanism. Use Figure 3.6 as a reference.

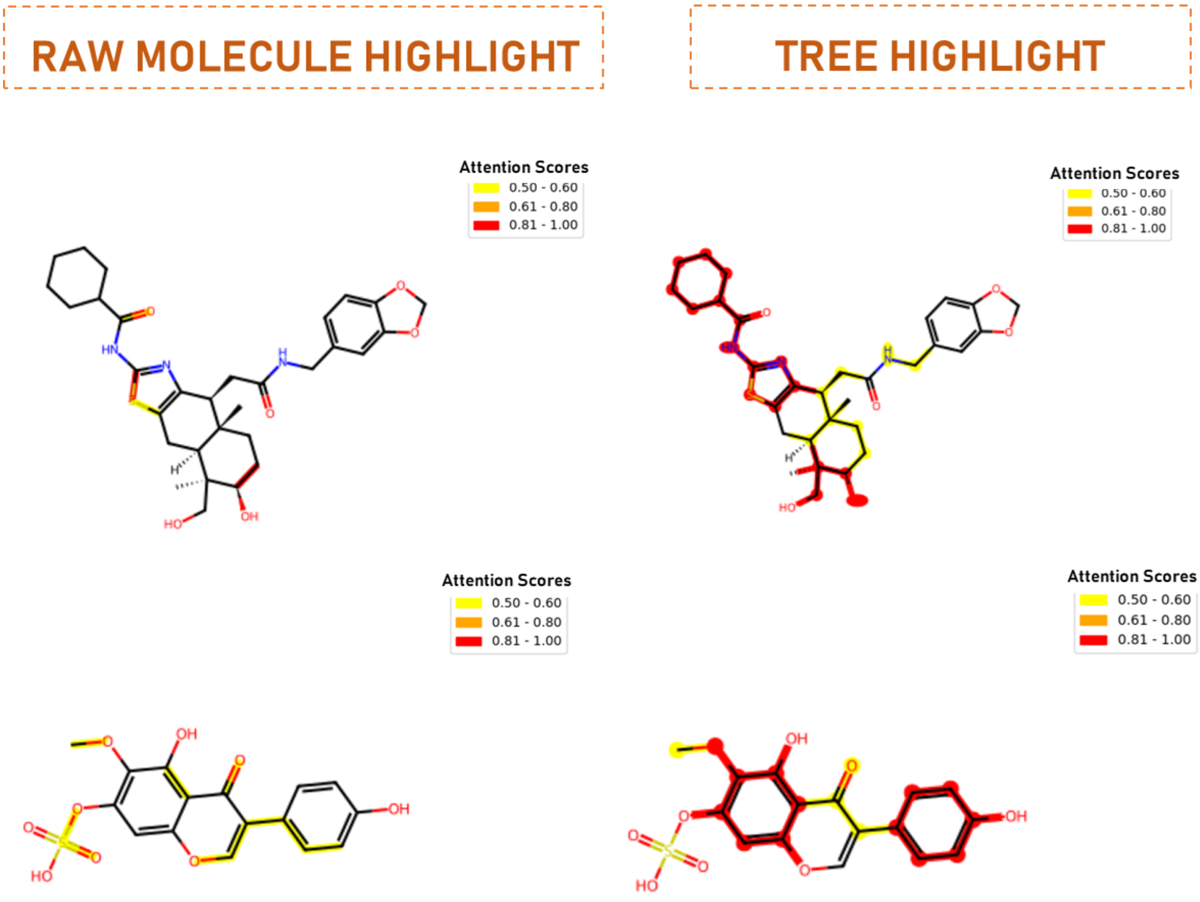


Figure 3.6: Here is an example illustrating the difference between the highlight functionality applied using attention scores from the processing of raw molecules and the one executed using attention scores from tree structure analysis. The right solution allows the researcher to highlight whole parts of the original molecule rather than individual bonds or atoms. A predefined color scheme empirically decided is then applied to represent different ranges of attention scores:
Low Attention (White): Bonds with normalized scores below 0.50 are colored white, indicating lower significance.
Medium Attention (Yellow to Orange): Bonds with scores between 0.50 and 0.80 are assigned colors ranging from yellow to orange, highlighting moderate importance.
High Attention (Red): Bonds with scores above 0.80 are colored red, signifying high importance and strong influence on the model's predictions.

Benefits of the Attention-Based Highlighting

The attention-based highlighting mechanism significantly enhances the interpretability and utility of machine learning models in chemical analysis. Integrating attention mechanisms with molecular visualization provides a powerful tool for elucidating the intricate relationships within chemical structures. Mapping attention scores to specific bonds and highlighting them based on their significance not only improves model transparency but also offers actionable insights into the fundamental properties of molecules. The introduction of the `deep_search` modality further refines this process by aggregating closely related high-attention bonds into larger, coherent fragments. This aggregation reduces visual distraction and maintains the structural integrity of significant molecular regions, resulting in more meaningful and comprehensive visualizations. This methodology bridges the gap between computational predictions and chemical intuition, facilitating a more informed analysis of molecular data. Applications in drug discovery and materials science can particularly benefit from this framework, as it enables enhanced analysis of chemical properties and behaviors. Additionally, color-coded visualizations ensure that complex attention data is presented immediately and intuitively, accessible even to those without a technical background in machine learning.

3.5 Steps Towards Interpretation

In this section, we outline the post-processing steps applied to the drug-like molecules identified by the model. After training and validating the model on the ZINC15 dataset, we applied it to molecules from the TCM dataset TM-MC2.0 to explore its predictions in the context of TCM. Here, we describe the steps taken to further analyze the TCM molecules classified as drug-like, leading to a deeper understanding of the results obtained.

The post-processing steps involve transforming molecular representations into lower-dimensional spaces, applying clustering techniques, and selecting optimal configurations based on performance assessments. This pipeline enabled in-depth exploration of drug-like molecular patterns through data-driven cluster analysis and led to the validation of the results provided by the model on TCM compounds.

3.5.1 Molecule Representation

In this study, each molecule identified as drug-like by the model was transformed into a high-dimensional fingerprint representation to enable effective clustering and analysis. This fingerprint was designed to quantitatively capture the molecular structure based on unique relevant fragment occurrences, providing a consistent format for subsequent dimensionality reduction and clustering.

The process began by identifying all unique relevant fragments from the set of drug-like molecules. These fragments were extracted using the model's predictions and the Attention Scores from the Transformer heads, which highlighted structural components characteristic of drug-like properties. As previously cited, we have empirically considered each fragment with a normalized Attention Score equal to or higher than 0.80. Each unique fragment was then assigned a specific dimension in the fingerprint vector. This resulted in a fingerprint length equal to the total number of unique relevant fragments, ensuring that each molecule could be represented in the same high-dimensional space.

For each molecule, the fingerprint vector was populated by counting the occurrences of each fragment within the molecular structure. If a specific fragment was present multiple times in a molecule, the corresponding element in the fingerprint vector reflected this count, as Figure 3.8 shows. This approach created a comprehensive profile of each molecule, capturing its structural features in a manner suitable for quantitative analysis.

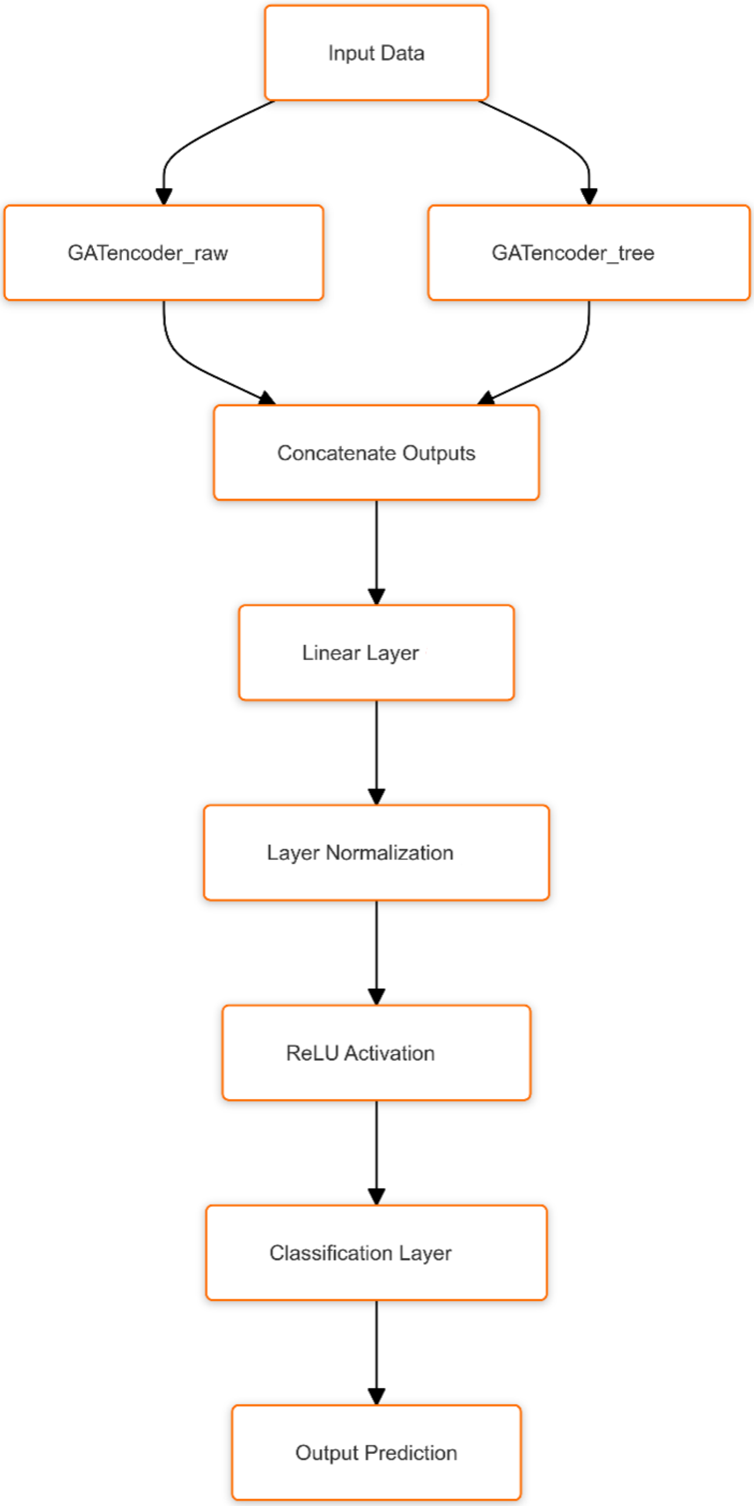


Figure 3.7: Representation of the entire model architecture. Input data is passed to the two Encoders, each of which returns an output. The two outputs are concatenated and pass through a Linear Layer and Normalization Layer, and after a ReLu Activation are passed to the MLP layer for final prediction.

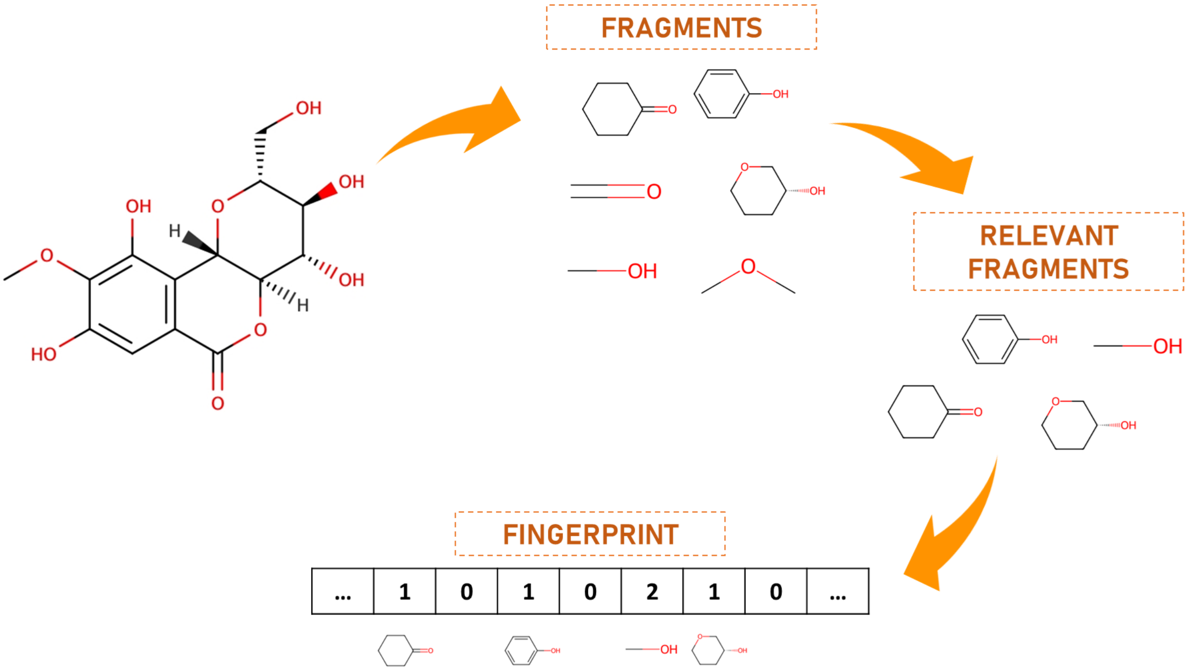


Figure 3.8: Schematic representation of the molecular encoding process. Starting with the input molecule, fragments are identified and extracted. Among these, relevant fragments are selected based on their Attention Scores. Finally, a fingerprint vector is generated, where each position corresponds to the occurrence of a specific relevant fragment in the molecule.

This fingerprint-based representation effectively translated complex molecular structures into a high-dimensional space where each dimension corresponded to a unique fragment. By providing a detailed and standardized format for all molecules, this representation enabled the consistent application of dimensionality reduction techniques, as well as reliable clustering based on structural similarities and differences among drug-like molecules.

3.5.2 Dimensionality Reduction

Dimensionality reduction was applied as a necessary preprocessing step to optimize clustering and computational efficiency. Each fingerprint vector representing a molecule had a length equal to the total number of unique fragments identified across all drug-like molecules, resulting in a multi-dimensional space with many sparse entries.

Data, if represented with a large number of dimensions, can create challenges for clustering and computational performance. This sparsity increases data storage requirements and computational costs while potentially introducing noise into downstream analysis. Additionally, the complexity of processing high-dimensional data can reduce the efficiency and scalability of the clustering algorithms.

Dimensionality reduction was used to transform the fingerprints into a lower-dimensional space, maintaining essential structural information while reducing redundancy. We evaluated multiple dimensionality reduction techniques, PCA [85] and UMAP [87], to achieve a balance between information preservation and dimensional simplicity. For additional technical details, see Subsection 2.6.1 and Subsection 2.6.2

This reduced representation enabled a more manageable and efficient data format, supporting clustering analysis by reducing the effects of sparsity and computational complexity.

PCA Application

To determine the optimal number of Principal Component (PC)s for reducing dimensionality, we used a combination of variance analysis and Scree Plot evaluation. The goal was to retain the minimum number of components necessary to capture the majority of the variance in the high-dimensional fingerprint data, ensuring that essential structural information was preserved while discarding noise and redundancy.

We generated a **Scree Plot** to visualize the variance explained by each component. This plot helped identify the point where additional components contributed diminishing returns to the total explained variance, commonly referred to as the **elbow point**. By selecting components that met a fixed threshold, we could ensure that the reduced representation contained sufficient structural detail from the original high-dimensional data. Once the optimal number of PCs was identified, clustering was applied to the reduced dataset.

UMAP Application

UMAP was chosen as an alternative dimensionality reduction method due to the high sparsity observed in the fingerprint data. Unlike PCA, which assumes linear relationships, UMAP is well-suited for capturing complex, non-linear structures in high-dimensional spaces, making it effective for this dataset's irregular density distribution [87].

To optimize UMAP, we experimented with multiple parameter configurations. Specifically, we adjusted three key parameters: `n_components`, `min_dist`, and `n_neighbors`. The `n_components` parameter, which specifies the number of dimensions in the reduced data, was set to retain an interpretable representation with a focus on balancing detail with dimensional simplicity. The `min_dist` parameter controls how closely UMAP clusters points together, influencing the compactness of clusters, while `n_neighbors` determines the size of the local neighborhood UMAP considers when mapping points in the lower-dimensional space.

Each parameter configuration was evaluated by applying clustering algorithms to the UMAP-reduced data. Multiple values for `n_components`, `min_dist`, and `n_neighbors` were tested to identify settings that yielded a cohesive and separable cluster structure. By adjusting these parameters, we aimed to capture the underlying structure of the fingerprint data while minimizing the impact of sparsity and noise.

3.5.3 Clustering

Clustering was employed to identify meaningful groupings within the drug-like molecules based on their fingerprint representations. This process aimed to uncover patterns and structural similarities among the molecules, enabling a more detailed exploration of the dataset.

Two clustering algorithms, **KMeans** and HDBSCAN (for technical details see Subsection 2.7.1 and 2.7.2), were applied to the reduced data obtained from dimensionality reduction techniques (PCA and UMAP). The quality of the clusters was assessed using Silhouette [103] and Density-Based Clustering Validation Index (DBCVI) [104] metrics. These methods were chosen for their complementary strengths: **KMeans** requires a predefined number of clusters and excels in partitioning data with distinct boundaries, while HDBSCAN dynamically identifies clusters of varying densities without requiring prior specification of the number of clusters. We ensured a comprehensive evaluation of the molecular collections by applying these methods to both the reduced versions of the dataset, PCA and UMAP.

The hierarchical framework we adopted allows HDBSCAN to detect clusters with varying densities and better capture the underlying data structure. It also reduces reliance on sensitive input parameters, such as the density threshold in DBSCAN, by shifting the focus to a more interpretable parameter that serves as both a smoothing factor and a cluster size threshold. To test various configurations of the algorithm, we varied the values bound to the hyperparameters

min_cluster_size and **min_samples** [105]. For KMeans, we tried different values related to the number of clusters to be generated, modifying the hyperparameter **n_clusters** [106].

The clustering results provided insight into the structural organization of drug-like molecules, serving as the basis for further analysis of cluster characteristics and relationships.

In this chapter, the methodologies and materials employed in this study were presented in detail. The processes of data preprocessing, feature extraction, and model architecture design were outlined, highlighting their significance in achieving robust and accurate results. Advanced methods, such as graph-based representations and deep learning frameworks, were utilized to address the challenges posed by the complexity of the dataset. These foundational steps ensure a solid basis for the results and discussions presented in the following chapters.

Chapter 4

Results and Discussion

This chapter presents the key findings of the study, focusing on the analysis of drug-like molecules identified by the model. The results include the model's performance on the ZINC dataset, which was used for training, validation, and testing. We then tried the ready-to-use model for the TM-MC2.0 dataset. Consequently, we report the results on molecule predictions, fragment extraction, dimensionality reduction, clustering, and insights into structural patterns within the dataset. These findings are supported by quantitative metrics, visualizations, and a comparative evaluation of the methods used.

4.1 Computational Architecture

The computational experiments were carried out using two distinct architectures, each suited to different phases of the workflow. For all tasks up to the post-processing stage, computations were performed on an office workstation featuring an Intel Core i7-10700 CPU @ 2.90 GHz with 16 cores, 16 GB of RAM, and an NVIDIA GeForce RTX 3060 GPU. The system ran on NVIDIA driver version 552.22 with CUDA version 12.4, providing adequate computational power for preprocessing, feature extraction, and model initialization.

A high-performance server was utilized for the post-processing stage. This server was equipped with an Intel Xeon Silver 4216 CPU with 64 cores (2 sockets, each with 16 cores and 2 threads per core), 256 GB of RAM, and an NVIDIA RTX A5000 GPU with 24 GB of memory. It ran on NVIDIA driver version 555.42.06 with CUDA version 12.5, delivering the computational capacity required for handling the intensive demands of these tasks.

4.2 Training and Testing Evaluation Metrics

To conduct the model hyperparameters fine-tuning and to comprehensively assess the classification performance of the trained model, we employed several standard evaluation metrics, including Accuracy, Precision, Recall, F1-score, and the Area Under the Receiver Operating Characteristic Curve (ROC-AUC). Below, we briefly define each metric:

- **Accuracy**: The proportion of correct predictions over the total number of predictions, calculated as:
$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{4.1}$$
where TP, TN, FP, and FN represent True Positives, True Negatives, False Positives, and False Negatives, respectively.

- **Precision**: The fraction of true positive predictions among all predicted positives, defined as:
$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4.2}$$

- **Recall (Sensitivity)**: The fraction of true positive predictions out of all actual positives:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4.3}$$

- **F1-Score**: The harmonic mean of Precision and Recall, balancing the two metrics:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.4}$$

- **ROC-AUC**: The ROC-AUC evaluates the model's ability to distinguish between classes across various decision thresholds.

  To construct the Receiver Operating Characteristic (ROC) curve, the following steps are followed:

  1. The model predicts probabilities for the positive class on the test set.
  2. Different thresholds are applied to convert probabilities into binary predictions.
  3. For each threshold, the True Positive Rate (TPR) and False Positive Rate (FPR) are calculated:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad \text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{4.5}$$

  4. The TPR and FPR values are plotted to form the curve.

  The Area Under the Curve (AUC) is computed as the integral of the area below the ROC curve, often using numerical methods such as the trapezoidal rule. A higher AUC indicates better discrimination performance.

These metrics provide complementary insights into the model's performance, ensuring a detailed evaluation of its classification ability.

4.3 Model Hyperparameters Tuning

As the first step of the model training, we had to find the optimal hyperparameter configuration to apply to the model. To achieve this, a subset of the ZINC15 training set, consisting of exactly 30,001 molecules, was used for the tuning process. Meanwhile, the test set originally designated for the model was repurposed as the validation set during this phase. We used Optuna [107] for this aim.

Optuna is an automated hyperparameter optimization framework that utilizes Bayesian optimization to identify the optimal hyperparameters for machine learning models testing the configurations through several study trials. It operates by defining one or more objective functions that assess a model's performance based on a given set of hyperparameters. Using these objective functions, Optuna efficiently navigates the hyperparameter space to find the best configurations.

The process involved several essential steps. First, we defined an objective function that specifies the hyperparameters to be optimized, using distributions provided by Optuna. This function ensures that the current parameters have not been previously tested, preventing redundancy, and then trains the model with the specified parameters, returning the evaluation metrics. We implemented a multi-objective function to optimize hyperparameter configurations effectively. Specifically, we evaluated configurations based on the ROC-AUC and loss function values, calculated by validating the model on the validation set every 5 epochs. The goal was to maximize the ROC-AUC while minimizing the loss. In addition, we ensured that the results of each trial were recorded for further analysis. The model was trained using 150 distinct random

Table 4.1: Hyperparameter Tuning Overview

Parameter	**Description**	**Tested Values**
batch_size	Determines the number of samples processed before the model is updated. Affects training speed and stability. Smaller batches improve gradient estimates but increase computational overhead.	16, 32, 64, 128.
learning_rate	Step size for updating the model weights in response to the estimated error. Controls the training progression. Too high can lead to divergence; too low may slow convergence.	Uniform distribution from 0.0001 to 0.1.
weight_decay	Adds a penalty to the loss function to prevent overfitting by discouraging large weights.	Uniform distribution from 0.00001 to 0.01.
sgd_momentum	Accelerates gradient vectors for faster convergence.	Uniform distribution from 0.8 to 0.99.
scheduler_gamma	Used in learning rate scheduling to reduce the rate by a factor of gamma.	Uniform distribution from 0.8 to 0.99.
pos_weight	Handles class imbalance by assigning more weight to the positive class in the loss function.	0.5, 0.7, 0.9, 1.0, 1.3, 1.5.
model_embedding_size	Determines the size of the embedding vectors in the model, affecting the capacity to capture features.	64, 128, 256, 512.
model_attention_heads	Specifies the number of attention heads, controlling the ability to capture multidimensional relationships.	1, 2, 3, 4, 5.
model_layers	Determines the number of layers, influencing the depth and capacity of the model.	1, 2, 3, 4, 5.
model_dropout_rate	Regularization technique to prevent overfitting by randomly deactivating units during training.	Uniform distribution from 0.2 to 0.8.
model_top_k_ratio	Specifies the ratio of nodes to retain in the top-k pooling layer, affecting the model's focus on key features.	Uniform distribution from 0.25 to 0.8.
model_top_k_every_n	Determines how frequently the top-k pooling layer is applied in the model.	1, 2, 3, 5.
model_dense_neurons	Specifies the number of neurons in the dense layers, impacting the model's learning capacity.	128, 256, 512.

configurations, each for 20 epochs, while tracking the loss and evaluation metrics computed for every configuration every 5 epochs. The best model from these evaluations was saved for each configuration. The objective function was integrated with Optuna to perform hyperparameter optimization, leveraging the results of the 150 trials and applying Bayesian optimization principles to maximize the ROC-AUC while minimizing the loss. The optimal results were saved and displayed after the optimization process. Additionally, if the option to load the state was enabled, the results of previous trials were loaded to continue the optimization from where it was left off. This configuration allowed us to efficiently explore the hyperparameter space and find the optimal combination for the model.

Table 4.2: Best Hyperparameter Configuration

Parameter	**Value**
`batch_size`	32
`learning_rate`	0.001186
`weight_decay`	0.000507
`sgd_momentum`	0.829522
`scheduler_gamma`	0.905972
`pos_weight`	0.7
`model_embedding_size`	256
`model_attention_heads`	3
`model_layers`	1
`model_dropout_rate`	0.222777
`model_top_k_ratio`	0.287710
`model_top_k_every_n`	1
`model_dense_neurons`	256

Through hyperparameter optimization, the configuration reported in Table 4.2 was identified as the best-performing setup for the model.

This configuration was identified as optimal based on model performance metrics, providing a balance between ROC-AUC and loss function values. The values for `learning_rate`, `weight_decay`, and `scheduler_gamma` were tuned to ensure stable training convergence, while structural parameters like `model_embedding_size` and `model_layers` supported the model's capacity to capture molecular relationships effectively.

4.4 Model Training and Testing on ZINC15

After identifying the optimal hyperparameters through the hyperparameter tuning process, the final model was trained using the complete training dataset from ZINC15 to fully exploit the available data. This training phase utilized the best hyperparameter configuration obtained from the tuning process, ensuring the model was optimized for generalization and performance. We utilized a training set comprising 206,827 samples (80% of the total), a validation set of 25,854 samples (10% of the total), and subsequently evaluated the trained model on a test set of 25,854 samples (10% of the total). The datasets maintained a balanced distribution between positive and negative classes, with each representing 50% of the total. We decided to train the model for up to 700 epochs; however, due to the Early stopping mechanism, ready to terminate the training if no improvement was observed within 125 consecutive epochs following the last recorded improvement, the training lasted just 325 epochs.

The performance of the trained model was evaluated by applying the model to the test set, using various metrics to ensure a comprehensive assessment of its classification ability. The adopted metrics are the same from Subsection 4.2 .

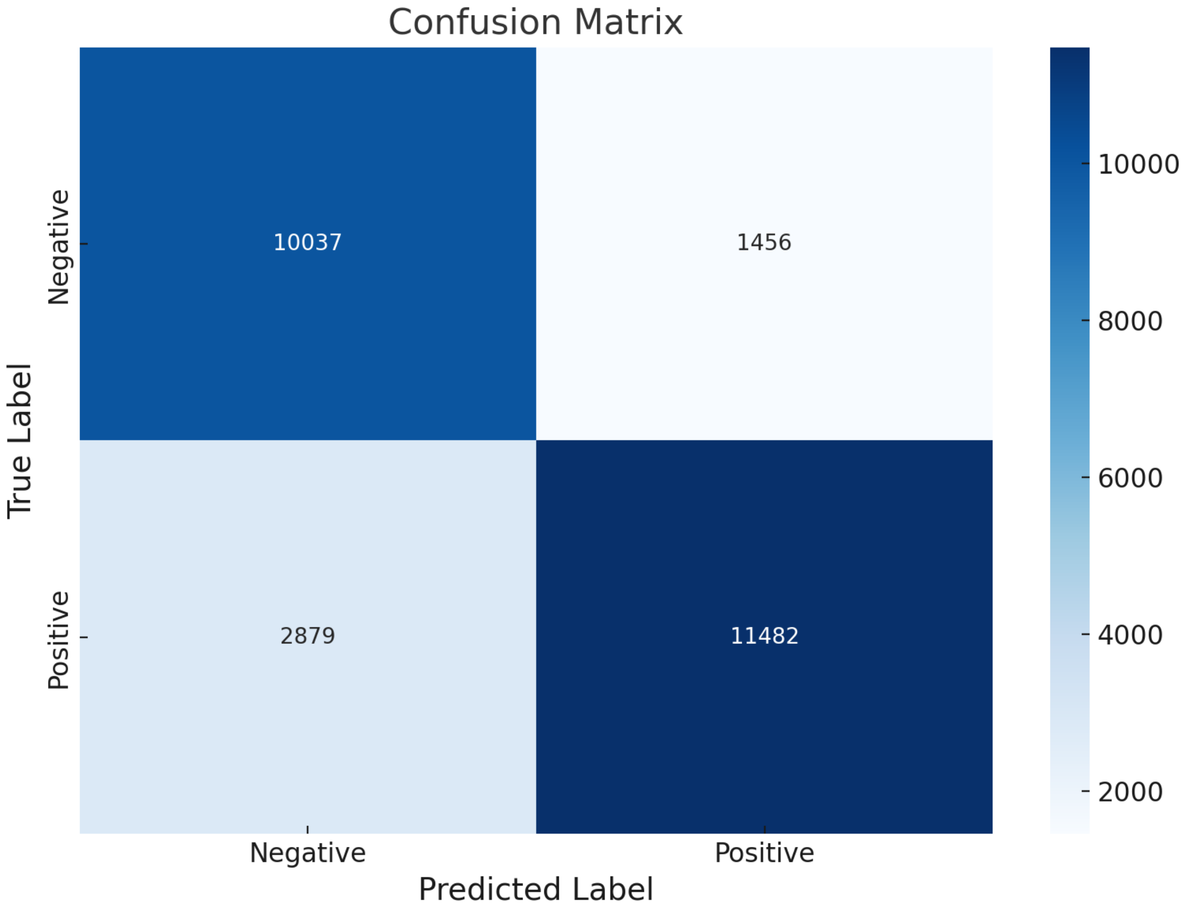


Figure 4.1: Confusion matrix illustrating the model's classification performance on the validation set. The matrix shows 10,037 True Negatives and 11,482 True Positives, representing correct predictions for negative and positive classes, respectively. Meanwhile, the model misclassified 2,879 samples as False Negatives and 1,456 samples as False Positives.

The confusion matrix based on this evaluation is illustrated in Figure 4.1, and the calculated evaluation metrics are displayed in Figure 4.2. Furthermore, Figure 4.3 presents a comparison of the metrics derived from the ZINC validation set and the test set.

The performance evaluation of the model on the ZINC15 test dataset, as summarized by the confusion matrix and various metrics, underscores its robustness and utility in drug discovery applications. The confusion matrix (Figure 4.1) reveals a strong classification capability, with 10,037 True Negatives and 11,482 True Positives accurately identified. These results are encouraging, as they demonstrate the model's ability to distinguish between in-vivo and in-vitro compounds effectively. However, the presence of 2,879 False Negatives and 1,456 False Positives indicates areas where further optimization could improve outcomes.

The high recall value of 88.75% (Figure 4.2 for a complete overview) is particularly meaningful in the context of drug discovery, where identifying as many potential active compounds as possible is critical. This metric ensures that the model is capturing a broad spectrum of candidates, minimizing the risk of prematurely excluding promising molecules. The slight trade-off observed in precision (79.95%) reflects a moderate rate of False Positives. However, in the context of drug discovery pipelines, false positives are typically regarded as a more acceptable outcome than false negatives. This is because false negatives represent missed opportunities for potential breakthroughs. Although having too many false positives in further pipeline steps can be very expensive, it's better to avoid having an excessively high rate of false

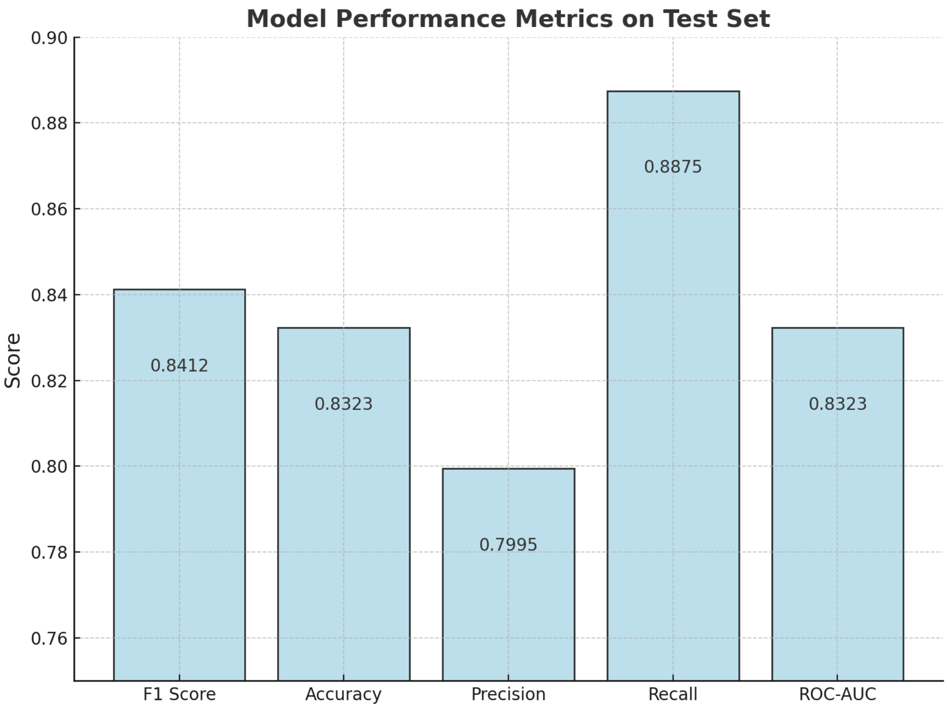


Figure 4.2: Bar plot of key performance metrics for the model when applied to the ZINC15 test dataset. The plot highlights the F1 Score (0.8412), Accuracy (83.23%), Precision (79.95%), Recall (88.75%), and ROC-AUC (0.8323). The high recall value indicates the model's strong ability to identify positive cases, while the precision reflects a moderate rate of false positives.

negatives [108] [109] [110]. These metrics collectively contribute to a high F1 score of 0.8412, indicating a well-balanced performance that aligns with the objectives of the study.

The ROC-AUC score of 0.8323 further validates the model's ability to distinguish between active and inactive molecules, indicating that it performs well across varying classification thresholds. This is particularly valuable for adapting the model to different stages of the drug discovery pipeline, where the importance of precision versus recall may vary. For example, in the early stages, high recall may be prioritized, while precision may become more critical during experimental validation to reduce costs.

When comparing the validation and test metrics (Figure 4.3), the consistency observed across the datasets demonstrates the model's generalization capability. Slight variations in recall and precision between these datasets suggest some level of data-specific behavior, which could be addressed by further diversifying the training dataset or applying techniques such as cross-validation to enhance robustness.

4.5 Application of the Validated Model to TM-MC2.0

After successfully training and validating the model on the ZINC15 dataset to establish its capacity to capture the general concept of drug-likeness, and evaluating its performance on this set of diverse molecules, we can confidently proceed to apply the model to TCM compounds. This transition enables us to assess the model's ability to generalize and predict drug-likeness within the specific chemical space represented by TCM, demonstrating its potential in real-world applications.

We predicted the labels of 20,974 compounds derived from the TM-MC2.0 dataset. Furthermore, for each molecule classified as drug-like, we identified and extracted the corresponding relevant fragments, facilitating their detailed analysis in subsequent steps.

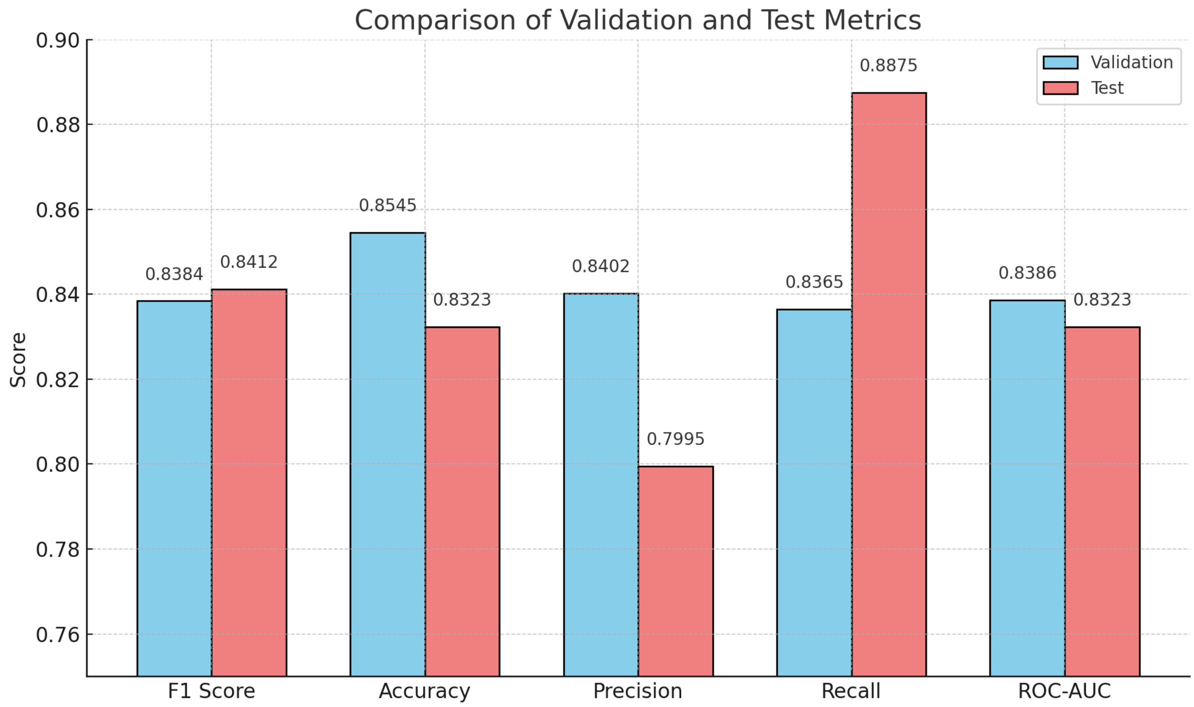


Figure 4.3: Comparison of model performance metrics between the validation and test datasets. The metrics include F1 Score, Accuracy, Precision, Recall, and ROC-AUC. The results highlight the consistency of the model's performance across both datasets, with slight variations observed in Recall and Precision.

From the total amount of compounds, 13,303 molecules have been predicted as drug-like.

4.6 Dimensionality Reduction and Clustering

Given the substantial number of molecules predicted as drug-like, in addition to the observation of significant structural diversity upon visualization, we decided to implement clustering techniques. This approach allows us to categorize the predicted drug-like molecules into distinct clusters, extract their centroids, and perform detailed analyses on the representative centroids of each cluster. This strategy facilitates a more structured evaluation of molecular diversity and enables insightful conclusions about the key structural features defining each cluster.

Before proceeding, it was essential to represent the molecules using a suitable fingerprinting method. To achieve this, we exploited the relevant fragments identified during the prediction step, constructing a fingerprint for each molecule based on the presence of these fragments. This approach, detailed in Subsection 3.5.1 and graphically shown in Figure 3.8, ensures a consistent and rich representation of the molecules, enabling effective methods of clustering and analysis.

However, the large number of unique relevant fragments extracted, specifically 1012, resulted in fingerprints that were highly sparse.

This low-density representation not only led to inefficient use of computational resources but also reduced the effectiveness of capturing and conveying meaningful molecular information. To ensure this, we will present a heat map reporting the density of information for each fingerprint. See Figure 4.4 for further details.

Due to the observed sparsity of the fingerprints, indicating a low density of information, we applied various data reduction techniques, specifically PCA and UMAP. A detailed examination of this topic can be found in the overview provided in Subsection 3.5.2. Each reduction method was paired with a different clustering algorithm, and the quality of the resulting clusters was

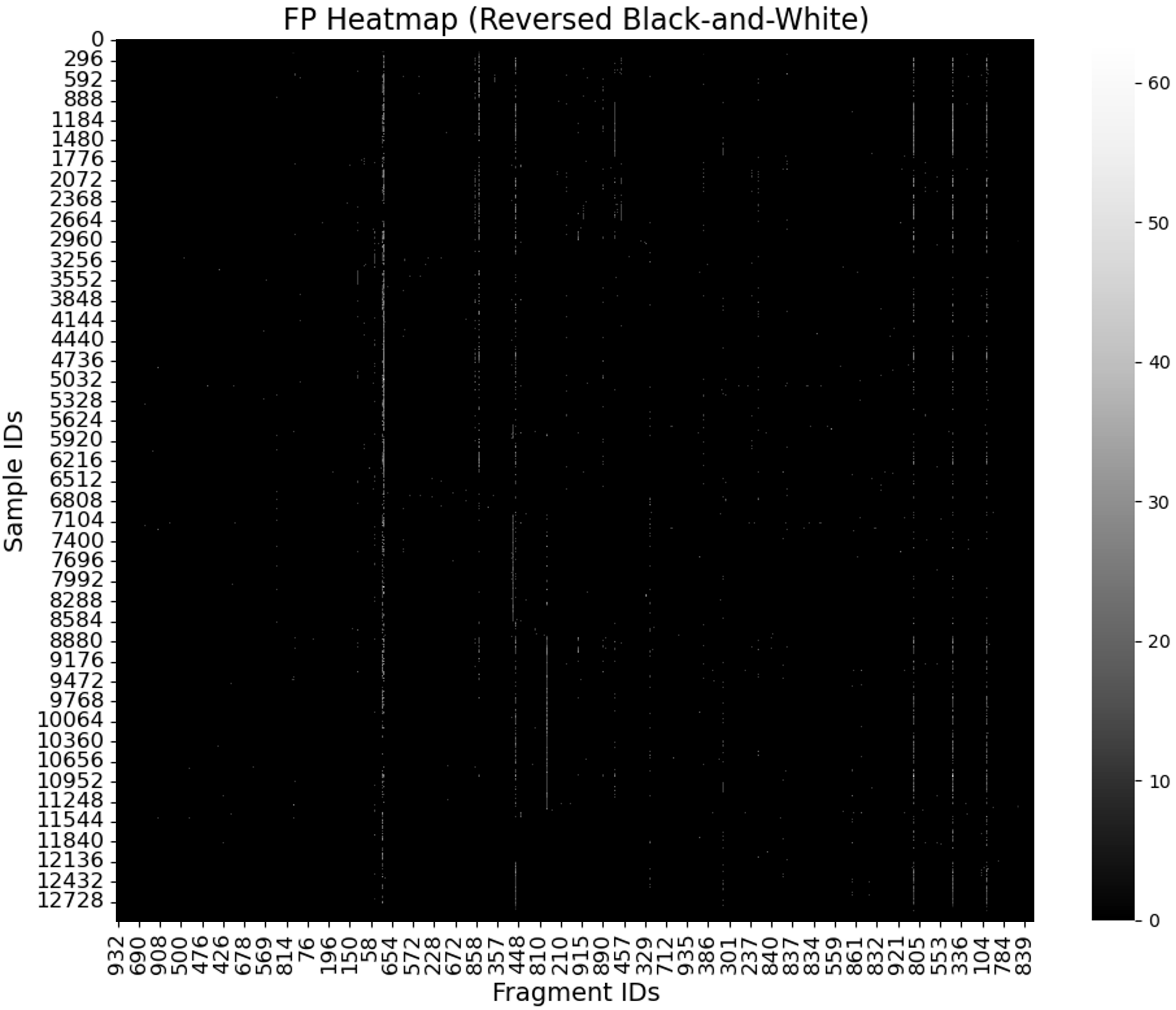


Figure 4.4: Heatmap illustrating the sparsity of molecular fingerprints, where each row represents a molecule and each column corresponds to a unique relevant fragment. The low density of non-zero values underscores the high sparsity in these fingerprint representations, emphasizing the importance of employing data reduction techniques for more effective analysis.

assessed using the Silhouette metric, as described in Subsection 3.5.3. Additionally, the DBCVI metric was used for internally selecting the optimal configuration of the HDBSCAN algorithm.

To determine the optimal number of PCs for data reduction using PCA, we used a **Scree Plot** to project the explained variance and visualize how the variance was distributed across the PCs of the model (see Figure 4.5). We found that to conserve at least 90% of the variance, 610 PCs must be used. We therefore proceeded by reducing in this way.

The UMAP technology works differently than PCA: instead of directly specifying the number of components you want to generate, it is necessary to set specific hyperparameters that affect how those components specifically are generated. The model then uses this information

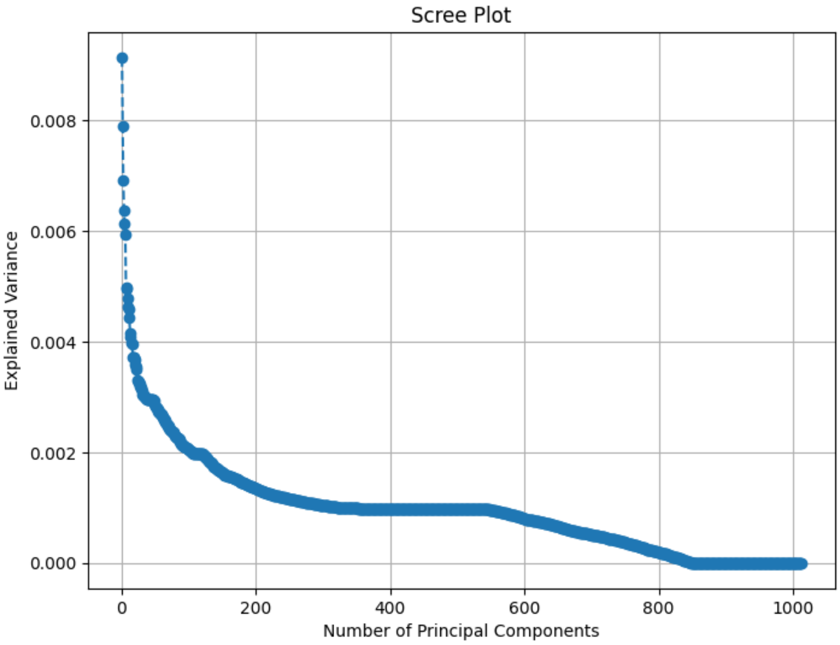


Figure 4.5: The Scree Plot illustrates the explained variance for each principal component derived from the PCA analysis. The x-axis represents the number of principal components, while the y-axis shows the proportion of variance explained by each component. The plot helps to determine the number of principal components required to explain a significant portion of the total variance in the dataset.

to generate the new reduced dimensionality encoding (see Subsubsection UMAP from Section 3.5.2). The hyperparameters tested in the list of combinations are reported in Table 4.3. Since there is no dedicated way to evaluate which hyperparameter configurations are optimal for UMAP, one must proceed with the clustering operation to make considerations. The values we tried for the clustering algorithms are the ones reported in Table 4.4. We used the Silhouette score to identify the optimal configuration, focusing only on combinations that achieved at least 80% coverage of samples within clusters, effectively minimizing the number of outliers to less than 20%. However, relying only on the Silhouette score was insufficient to determine the ideal configuration for UMAP, as many configurations produced similar Silhouette values. Therefore, we introduced a second metric, DBCVI, to further refine the selection process and ensure a more accurate identification of the optimal clustering setup. Since DBCVI relies on the concept of cluster density, it is particularly indicated for clustering methods such as HDBSCAN [104]. In the end, we averaged the Silhouette and DBCVI scores for all configurations that achieved at least 80% coverage and selected the configuration with higher mean reported.

In conclusion, the configuration ultimately selected employs UMAP as the dimensionality reduction technique, as detailed in Table 4.5, integrating UMAP with HDBSCAN. In contrast, KMeans clustering, discussed in Subsubsection 2.7.1, aim to minimize variance within clusters but with more dispersed and less densely grouped. This is reflected in the lower DBCVI, even though the method achieved a higher Silhouette score.

4.6.1 Clustering Configuration Evaluation

As clustering techniques, we applied **KMeans** and HDBSCAN to the reduced data and subsequently evaluated the quality of the resulting clusters using appropriate metrics.

Table 4.3: Hyperparameters Tested for UMAP

Parameter Name	**Ranges**	**Description**
n_neighbors	10, 15, 20, 25, 30, 35, 40, 45	Controls the size of the local neighborhood used for manifold approximation in UMAP
min_dist	0.0, 0.1, 0.2, 0.3, 0.5, 0.8, 0.9	The minimum distance between points in UMAP, influencing cluster compactness
n_components	20, 25, 30, 35, 40, 45, 50	The dimensionality of the reduced space in UMAP

To validate the performance of **KMeans**, a range of potential hyperparameter configurations was defined, as detailed in Table 4.4. For each configuration, the Silhouette score was computed and recorded for subsequent analysis. Similarly, for HDBSCAN, the parameter ranges for **min_samples** and **min_cluster_size** were defined, as shown in Table 4.4.

Table 4.4: Clustering Parameters for **KMeans** and HDBSCAN Validation

Clustering Method	**Parameter Name**	**Ranges**	**Parameter Meaning**
KMeans	*n_clusters_list*	10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 125, 150, 175, 200, 250, 300	Number of clusters to partition the dataset into
HDBSCAN	*min_samples*	5, 10, 20	Minimum number of samples in a neighborhood for a point to be considered a core point
	min_cluster_size	30, 50, 70, 100, 150	Minimum number of points to form a cluster

Silhouette scores were calculated and recorded, and the results were analyzed to determine the most effective data reduction technique based on the highest Silhouette score.

The analysis demonstrated that the best-performing configurations using UMAP consistently and significantly outperformed all configurations tested with PCA, irrespective of the clustering method employed, as displayed in Figure 4.6.

Given that the UMAP configurations exhibited comparable Silhouette scores, we leveraged the density-based clustering focus of HDBSCAN, utilizing the DBCVI metric to identify the optimal configuration.

Ultimately, we determined that the combination of HDBSCAN with UMAP proved to be the most promising approach since it presented the highest average between Silhouette and DBCVI (see Figure 4.7). Consequently, we proceeded with this method, utilizing the configuration detailed in Table 4.5.

4.7 Centroid Extraction and Bibliographic Validation for Comprehensive Pharmacological Interpretation

In this section, we describe the process of extracting centroids from each cluster and performing a detailed bibliographic review of both the medoid and the molecules closest to these medoids. The main difference between a centroid and a medoid is that a centroid is the mathematical representation of the central point within a cluster, even if that point is not a member of the

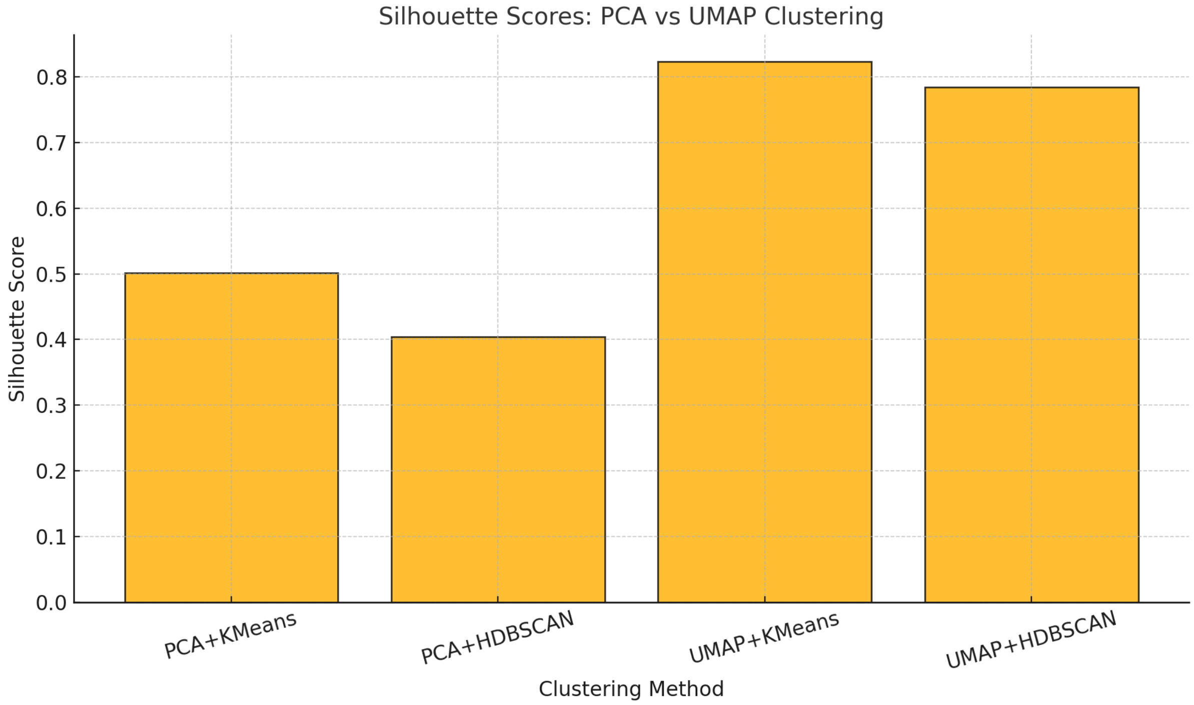


Figure 4.6: For each configuration, we reported only the one achieving the highest Silhouette score. The objective is to identify the optimal configuration for each combination of dimensionality reduction and clustering methods. It is evident that UMAP performs significantly better than PCA, at least in this particular case.

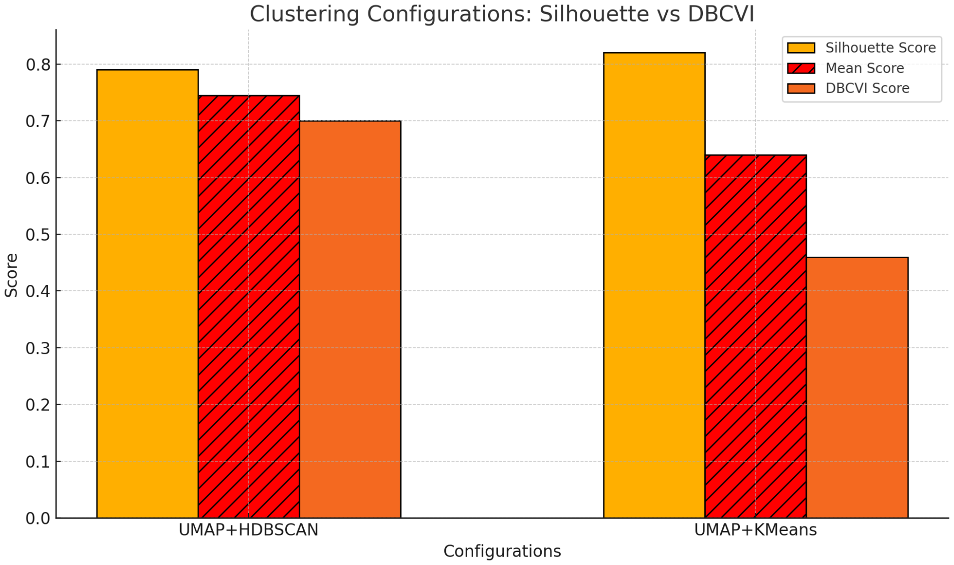


Figure 4.7: We conducted a comparative analysis of the Silhouette score and DBCVI for configurations with higher mean values across the two metrics obtained using UMAP in conjunction with both KMeans and HDBSCAN. Based on the results, we opted for the UMAP-HDBSCAN combination, as it demonstrated a superior average performance across the two metrics.

Table 4.5: Best Clustering Results Configuration: UMAP + HDBSCAN

num neighbors	**min dist**	**num components**	**min samples**	**min cluster_size**	**num clusters_found**	**% clustered**	**Silhouette**	DBCVI
15	0.0	35	20	30	147	85.4845	0.7837	0.7054

cluster. The medoid is the actual element within the cluster that is closest to the mathematical centroid. This means that it has the lowest sum of dissimilarities to all other objects in the cluster [111]. If we want to seek real bibliographical information about the molecules that are the centers of the clusters, we have to ensure that the embeddings correspond to real molecules in the dataset. Therefore, we prefer medoids instead of centroids.

The goal was to identify key chemical features and validate the drug-likeness of the identified compounds by connecting them with known pharmacological data. This process enabled us to gain a better understanding of the therapeutic potential within each cluster and to assess the effectiveness of the model in identifying meaningful molecular patterns.

After applying the clustering technique with the parameters specified in Table 4.5, we extracted the medoids for each cluster, i.e. the molecule strictly present in the dataset closest to the mathematical centroid of each cluster. This allowed us to extract a representative for each cluster, summarizing the main properties of the molecules assigned to it. This procedure was crucial for reducing the number of elements requiring bibliographic research, thereby validating the capabilities of the trained model. Using the same distance metric operated for the HDBSCAN clustering operation, i.e. the Euclidean distance, we subsequently identified the molecule closest to each medoid. This provided two molecules per cluster for which to seek information, in case the data obtained for the medoid were inconsistent or inconclusive for our research purposes. The primary sources of information are listed in Table 4.6.

Table 4.6: Sources and Databases Used for Clusters' Bibliographic Research

Name	**Description**
Natural Product Activity and Species Source (NPASS) [112]	Chemical Structure Database
National Library of Medicine's Medical Subject Headings (MeSH) [113]	Vocabulary Thesaurus
PubChem [78]	Chemical Database
PubMed [114]	Biomedical Literature
Chemical Entities of Biological Interest (CHEBI) [74]	Chemical Structure Database
ChemSpider [115]	Chemical Structure Database
Chemical European Molecular Biology Laboratory (ChEMBL) [75]	Chemical Structure Database
Cymitquimica [116], Ambeed [117], BLDPharm [118], BioSynth [119]	Online Drug Marketplaces
ELSEVIER [120]	Publisher
Japan Chemical Substance Dictionary [121]	Biomedical Literature
TM-MC.kr [13]	Online Resource
pub.acs.org [122]	American Chemical Society
DrugBank [79]	Drug Information Database
Google Patents [123]	Patent Search Engine

From this analysis, it emerged that out of the 147 clusters identified, 112 were classified as relevant to the pharmaceutical field. Specifically, for a cluster to be considered pharmaceutical, at least one molecule, either the medoid or its closest sample, needed to have demonstrated pharmaceutical use or to already be an approved drug. In 17 clusters, further experiments are required to clarify their potential. For 17 clusters, no relevant information was found, while for one cluster, clear information indicated its industrial use with limited pharmaceutical applicability (more details in the Supplementary Table).

The following paragraphs present four illustrative examples of clusters analyzed to provide further clarification regarding the analytical method employed.

Cluster 39 - Corycavamine and Coptisine Both the medoid and its closest molecule have been confirmed as drugs, with both demonstrated to be effective allies in combating Alzheimer's Disease (AD).

Finding sufficient information in their case was remarkably straightforward: they exemplify a clear potential as anti-Alzheimer agents, supported by abundant sources. For Corycavamine, the dedicated PubChem page [124] lists several studies confirming its role as a BACE1 inhibitor, indicating the potential for AD treatment. Indeed, PubChem has consistently served as our primary starting point for research, as it is arguably the most comprehensive source of information and integrates research from a multitude of sources. From this point, a link to the MeSH website reveals that MeSH categorizes the molecule in a similar manner [125]. For an alternative perspective, we explored Cymitquimica's drug marketplace [126], which instead emphasizes its role as an anti-inflammatory agent in TCM. Hence, we can confirm that Corycavamine is a potential drug candidate, validating the accuracy of our model's prediction.

About Coptisine, it was recently identified as a potential contributor to the pathogenesis of AD. It is a key pharmacologically active component of the TCM prescription Oren-gedoku-to, which holds therapeutic promise for AD treatment. This hypothesis is supported by multiple sources available on the corresponding PubChem page [127], with one paper, in particular, providing strong confirmation [128].

Cluster 84 - Vinaginsenoside R17 and Notoginsenoside R1 Analyzing Cluster 84 presented some challenges. Specifically, the medoid SMILES did not have a corresponding entry in PubChem. To resolve this, we consulted alternative sources and used the TM-MC 2.0 website to identify the molecule's name, ultimately determining it to be Vinaginsenoside R17 [129]. Subsequently, we identified its corresponding page on PubChem and confirmed that its 2D structure and chemical formula were consistent. However, further investigation revealed that no useful information was available, either online or on ChemSpider, which we utilized as an alternative resource to PubChem and to verify the presence of drugs in online marketplaces. Therefore, confirming the prediction for that drug was not possible due to the lack of available resources.

For the other molecule, we repeated the initial step applied to the previous molecule and identified its name as Notoginsenoside R1. In this case, the amount of available information was exceptional. PubChem describes this molecule as an antioxidant, an apoptosis inducer, a phytoestrogen, and a neuroprotective agent in its "Use and Manufacturing" section [130]. Additionally, through the "Information Sources" section, we accessed other databases such as CHEBI [131] and ChEMBL [132], which corroborated these evaluations.

Cluster 70 - Malonyl-saikosaponin E and Malonyl-saikosaponin A There are instances where, despite searching through all the resources listed in Table 4.6, the information obtained was insufficient to evaluate the model's prediction. This is the case for the molecules in Cluster 70, for which we only found a single paper mentioning the beneficial properties of the plant containing these compounds, without providing any relevant details about the specific molecules themselves [133]. Such a situation might indicate that the model has identified potentially drug-like molecules that have yet to be thoroughly analyzed in the current scientific literature.

Cluster 15 - Myricetin 3'-xyloside and Quercetin-7-O-p-coumaroyl Glucoside In some clusters, the representative molecules provided useful but insufficient information to confirm or refute the model's predictions. For example, in Cluster 15, PubChem offered access to various details about the molecules, but nothing particularly impactful. For Myricetin 3'-xyloside, we accessed its PubChem page [134], which linked to the Japan Chemical Substance Dictionary [135]. This source referenced only one paper of interest, which highlighted potential antioxidant effects. However, there was a lack of clear evidence supporting its definitive efficacy [136]. Regarding the second molecule, no information was found. However, the fact that it represents a variant of the better-known and effective Quercetin [137] suggests that its properties might be worth further investigation.

Cluster 4 - Isoamyl butyrate and cis-3-Hexenyl isobutyrate Finally, we present a case where the analyzed molecules may represent a potential misprediction. This is the case for Cluster 4, which contains around 80 molecules in total, where the representative molecules exhibit clear toxic and harmful properties for humans. These molecules are typically fragrances, industrial chemicals, or cleaning agents. For example, PubChem, citing European authorities and the U.S. Environmental Protection Agency (EPA) specifically [138], highlights that Isoamyl butyrate is primarily used as a fruity fragrance or in household cleaning products [139]. The same applies to cis-3-Hexenyl isobutyrate as well [140].

Overall, these results confirm the reliability of the model in facilitating efficient and accurate molecular screening. By identifying active compounds with high sensitivity and maintaining good precision, the model has the potential to support the overarching goal of reducing the time and cost associated with traditional drug discovery methods.

Chapter 5

Conclusion

In this thesis, we managed the complex challenge of analyzing molecular structures using advanced GNN methodologies. We began by applying a Transformer GNN framework and training the model on our molecular dataset, specifically targeting the extraction of significant fragments from molecular structures. The trained model was subsequently applied to the TCM dataset, where we employed the model's attention mechanisms alongside a tree decomposition algorithm to identify and extract relevant fragments and predict the drug-likeness for the targeted molecules.

Using these extracted fragments, we constructed molecular fingerprints of the molecules predicted as drugs, which served as the basis for further analysis. An evaluation was conducted to determine the optimal combination of dimensionality reduction and clustering techniques with the objective of creating clusters from which medoids could be derived. By systematically evaluating dimensionality reduction and clustering combinations, we ensured an efficient and insightful exploration of the dataset. The medoids were subsequently used as references for further online research to expand our understanding of their potential biological properties and clinical application. Most of the medoids confirmed the predictions, demonstrating the accuracy of our model. However, for some, the academic information available was limited, indicating the need for further studies to validate the prediction. This evidence suggests a possible application of our model: using it upstream in the drug discovery process to simplify the selection of candidate molecules and accelerate the initial steps of the pipeline. For some cases, we confirmed instances of misprediction, highlighting that the model still requires refinement.

The results obtained confirm that our approach effectively leverages the power of GNNs and attention mechanisms to not only predict molecular properties but also provide valuable insights into the structural features driving these properties.

There are several promising directions for future work originating from this research. One critical aspect of advancing this research involves refining the dataset used for training. The current dataset is effective in distinguishing molecules based on in vitro and in vivo activity, providing a solid foundation. However, it lacks the granularity required for deeper insights. Expanding the dataset to classify drug-like versus non-drug-like molecules would enhance its applicability to real-world drug discovery. Furthermore, linking molecular data to the specific diseases these compounds are intended to treat, rather than focusing exclusively on their overall properties, could provide a more practical and clinically relevant perspective. This shift in focus would enable research to explore therapeutic contexts more directly and potentially reveal patterns that might otherwise be unseen.

Another valuable direction would be to focus on the unique properties of compounds in TCM, highlighting their traditional therapeutic functions and synergistic effects. These combinations, which are central to TCM, remain underexplored in the context of this study. Investigating why certain combinations are effective could validate traditional practices and uncover new opportunities for modern drug development.

Studying the chemical and biological properties of the extracted fragments concerning their pharmacological activities could enrich the interpretation of the model predictions. By aligning the research with these larger and more targeted objectives, the study can achieve a deeper understanding of the molecular space and its implications for drug discovery.

In conclusion, this research has demonstrated the potential of advanced GNN methodologies, particularly Transformer-based models, in effectively analyzing molecular structures and predicting drug-likeness with a focus on TCM. The results underscore the value of integrating attention mechanisms and fragment-based analysis to provide meaningful insights into molecular properties. Moving forward, expanding the dataset, exploring traditional compound combinations, and heightening the biological contextualization of extracted fragments present promising directions for enhancing the scope and impact of this work. By continuing along these lines, the study can contribute to more efficient and targeted drug discovery processes, ultimately advancing the field of computational chemistry and molecular analysis and also leading to a more comprehensive scientific understanding of TCM.

Bibliography

[1] Song Xuan Ke. The principles of health, illness and treatment - the key concepts from "the yellow emperor's classic of internal medicine". *Journal of Ayurveda and Integrative Medicine*, 14(1):100637, 2023. doi:10.1016/j.jaim.2022.100637.

[2] Ling Li, Lele Yang, Liuqing Yang, Chunrong He, Yuxin He, Liping Chen, Qin Dong, Huaiying Zhang, Shiyun Chen, and Peng Li. Network pharmacology: a bright guiding light on the way to explore the personalized precise medication of traditional chinese medicine. *Chinese Medicine*, 18:Article 146, 2023. doi:10.1186/s13020-023-00853-2.

[3] Britannica, T. Editors of Encyclopaedia. Traditional chinese medicine. https://www.britannica.com/science/traditional-Chinese-medicine, 2024. Encyclopedia Britannica, October 14, 2024.

[4] Lingru Li, Haiqiang Yao, Ji Wang, Yingshuai Li, and Qi Wang. The role of chinese medicine in health maintenance and disease prevention: Application of constitution theory. *American Journal of Chinese Medicine*, 47(3):495–506, 2019. doi:10.1142/S0192415X19500253.

[5] Hong Luo, Hong Chen, Chunhua Liu, et al. The key issues and development strategy of chinese classical formulas pharmaceutical preparations. *Chinese Medicine*, 16:70, 2021. doi:10.1186/s13020-021-00483-6.

[6] D.B. Singh, R.K. Pathak, and D. Rai. From traditional herbal medicine to rational drug discovery: Strategies, challenges, and future perspectives. *Revista Brasileira de Farmacognosia (Rev. Bras. Farmacogn.)*, 32:147–159, 2022. doi:10.1007/s43450-022-00235-z.

[7] Ann C. Marshall. Traditional chinese medicine and clinical pharmacology. In *Drug Discovery and Evaluation: Methods in Clinical Pharmacology*, page 455. Springer, March 2020. doi:10.1007/978-3-319-68864-0_60.

[8] Xin Sun, Ling Li, Yanmei Liu, Wen Wang, Minghong Yao, Jing Tan, Yan Ren, Ke Deng, Yu Ma, Yuning Wang, Jin Chen, Wei Huang, Qing Xia, Youping Li, and Hongcai Shang. Assessing clinical effects of traditional chinese medicine interventions: Moving beyond randomized controlled trials. *Frontiers in Pharmacology*, 12, 2021. doi:10.3389/fphar.2021.713071.

[9] Yinglian Song, Wanyue Chen, Ke Fu, and Zhang Wang. The application of pearls in traditional medicine of china and their chemical constituents, pharmacology, toxicology, and clinical research. *Frontiers in Pharmacology*, 13, 2022. doi:10.3389/fphar.2022.893229.

[10] Jian Wang, Mei-Yu Wu, Jian-Qiu Tan, et al. High content screening for drug discovery from traditional chinese medicine. *Chinese Medicine*, 14:5, 2019. doi:10.1186/s13020-019-0228-y.

[11] Xing-Xin Yang, Wen Gu, Li Liang, Hong-Li Yan, Yan-Fang Wang, Qian Bi, Ting Zhang, Jie Yu, and Gao-Xiong Rao. Screening for the bioactive constituents of traditional chinese medicines progress and challenges. *RSC Adv.*, 7:3089–3100, 2017. URL: http://dx.doi.org/10.1039/C6RA25765H, doi:10.1039/C6RA25765H.

[12] Qiang Lv, Guanhua Chen, Hui He, et al. TCMBank - the largest TCM database provides deep learning-based chinese-western medicine exclusion prediction. *Signal Transduction*

and Targeted Therapy, 8:127, 2023. doi:10.1038/s41392-023-01339-1.

[13] Sang-Kyun Kim, Myung-Ku Lee, Ho Jang, Jeong-Ju Lee, Sanghun Lee, Yunji Jang, Hyunchul Jang, and Anna Kim. TM-MC 2.0: an enhanced chemical database of medicinal materials in northeast asian traditional medicine. *BMC Complementary Medicine and Therapies*, 24:Article 40, 2024. doi:10.1186/s12906-024-040.

[14] Calvin Yu-Chian Chen. TCM Database@Taiwan: The World's Largest Traditional Chinese Medicine Database for Drug Screening In Silico. *PLoS ONE*, January 2011. doi:10.1371/journal.pone.0015939.

[15] Qi Ge, Liang Chen, Yi Yuan, Lanlan Liu, Fan Feng, Peng Lv, Shangshang Ma, Keping Chen, and Qin Yao. Network pharmacology-based dissection of the anti-diabetic mechanism of lobelia chinensis. *Frontiers in Pharmacology*, 11:347, 2020. doi:10.3389/fphar.2020.00347.

[16] Damilola S. Bodun, Damilola A. Omoboyowa, Olaposi I. Omotuyi, Ezekiel A. Olugbogi, Toheeb A. Balogun, Chiamaka J. Ezeh, and Emmanuel S. Omirin. Qsar-based virtual screening of traditional chinese medicine for the identification of mitotic kinesin eg5 inhibitors. *Computational Biology and Chemistry*, 104:107865, 2023. URL: https://www.sciencedirect.com/science/article/pii/S1476927123000567, doi:10.1016/j.compbiolchem.2023.107865.

[17] Shuang Ma, Jing Liu, Wei Li, et al. Machine learning in tcm with natural products and molecules: current status and future perspectives. *Chinese Medicine*, 18:43, 2023. doi:10.1186/s13020-023-00741-9.

[18] Teague Sterling and John J. Irwin. ZINC 15–Ligand Discovery for Everyone. *Journal of Chemical Information and Modeling*, 55(11):2324–2337, November 2015. doi:10.1021/acs.jcim.5b00559.

[19] Andrey A. Toropov and Alla P. Toropova. Application of smiles to cheminformatics and generation of optimum smiles descriptors using coral software. In Alla P. Toropova and Andrey A. Toropov, editors, *QSPR/QSAR Analysis Using SMILES and Quasi-SMILES*, volume 33 of *Challenges and Advances in Computational Chemistry and Physics*. Springer, Cham, 2023. doi:10.1007/978-3-031-28401-4_3.

[20] Jian Zhang, Li Chen, and Wei Zhou. Hybrid qsar and deep learning approach to address the complexity of multi-component systems in traditional chinese medicine. *Computational and Structural Biotechnology Journal*, 21:561–574, 2023. doi:10.1016/j.csbj.2023.03.014.

[21] Arindam Paul, Dipendra Jha, Reda Al-Bahrani, Wei keng Liao, Alok Choudhary, and Ankit Agrawal. Chemixnet: Mixed dnn architectures for predicting chemical properties using multiple molecular representations, 2018. URL: https://arxiv.org/abs/1811.08283, arXiv:1811.08283.

[22] Tianxiang Zhao, Xiang Zhang, and Suhang Wang. Graphsmote: Imbalanced node classification on graphs with graph neural networks. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining (WSDM)*, pages 833–841, 2021. doi:10.1145/3437963.3441720.

[23] Li Wang, Mei Zhao, and Qiang Liu. A database for traditional chinese medicine: Improving data quality for enhanced predictive modeling. *Journal of Ethnopharmacology*, 250:112469, 2020. doi:10.1016/j.jep.2020.112469.

[24] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020. URL: https://www.sciencedirect.com/science/article/pii/S2666651021000012, doi:10.1016/j.aiopen.2021.01.001.

[25] Dandi Jiang, Zhenpeng Wu, Ching-Yun Hsieh, et al. Could graph neural networks learn better molecular representation for drug discovery? a comparison study of descriptor-based and graph-based models. *Journal of Cheminformatics*, 13:12, 2021. doi:10.1186/

s13321-020-00479-8.

[26] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio', and Yoshua Bengio. Graph attention networks, 2018. URL: https://arxiv.org/abs/1710.10903, arXiv:1710.10903.

[27] Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs, 2021. URL: https://arxiv.org/abs/2012.09699, arXiv:2012.09699.

[28] Xian bin Ye, Quanlong Guan, Weiqi Luo, Liangda Fang, Zhao-Rong Lai, and Jun Wang. Molecular substructure graph attention network for molecular property identification in drug discovery. *Pattern Recognition*, 128:108659, 2022. URL: https://www.sciencedirect.com/science/article/pii/S0031320322001406, doi:10.1016/j.patcog.2022.108659.

[29] deepfindr. Gnn project. https://github.com/deepfindr/gnn-project, 2024. Accessed: November 2024.

[30] Kai Gao, WanChen Cao, ZiHao He, Liu Liu, JinCheng Guo, Lei Dong, Jini Song, Yang Wu, and Yi Zhao. Network medicine analysis for dissecting the therapeutic mechanism of consensus tcm formulae in treating hepatocellular carcinoma with different tcm syndromes. *Frontiers in Endocrinology*, 15, 2024. doi:10.3389/fendo.2024.1373054.

[31] Xinfeng Zhou, Sing Wai Seto, Dennis Chang, Hosen Kiat, Valentina Razmovski-Naumovski, Kelvin Chan, and Andrew Bensoussan. Synergistic effects of chinese herbal medicine: A comprehensive review of methodology and current research. *Frontiers in Pharmacology*, 7:201, July 2016. doi:10.3389/fphar.2016.00201.

[32] Mengmeng Wang, Fengting Yin, Ling Kong, Le Yang, Hui Sun, Ye Sun, Guangli Yan, Ying Han, and Xijun Wang. Chinmedomics: a potent tool for the evaluation of traditional chinese medicine efficacy and identification of its active components. *Chinese Medicine*, 19:Article 47, 2024. doi:10.1186/s13020-024-0047-0.

[33] Meng Zhao, Yanan Che, Yan Gao, and Xiangyang Zhang. Application of multi-omics in the study of traditional chinese medicine. *Frontiers in Pharmacology*, 15, 2024. doi:10.3389/fphar.2024.1431862.

[34] Revive by GARDP. Drug-likeness definition. https://revive.gardp.org/resource/druglikeness/?cf=encyclopaedia, 2024. Accessed: November 2024.

[35] Sheng Tian, Junmei Wang, Youyong Li, Dan Li, Lei Xu, and Tingjun Hou. The application of in silico drug-likeness predictions in pharmaceutical research. *Advanced Drug Delivery Reviews*, 87:102–113, 2015. doi:10.1016/j.addr.2015.01.009.

[36] Arup K. Ghose, Vellarkad N. Viswanadhan, and John J. Wendoloski. A knowledge-based approach in designing combinatorial or medicinal chemistry libraries for drug discovery. 1. a qualitative and quantitative characterization of known drug databases. *Journal of Combinatorial Chemistry*, 1(1), December 1998. doi:10.1021/cc9800071.

[37] Christopher A. Lipinski, Francesca Lombardo, Beryl W. Dominy, and Paul J. Feeney. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Advanced Drug Delivery Reviews*, 46(1–3):3–26, January 1997. doi:10.1016/S0169-409X(00)00129-0.

[38] G. Richard Bickerton, Gaia V. Paolini, Jérémy Besnard, Sorel Muresan, and Andrew L. Hopkins. Quantifying the chemical beauty of drugs. *Nature Chemistry*, 4:90–98, January 2012. doi:10.1038/nchem.1243.

[39] Ana Laura Dias, Latimah Bustillo, and Tiago Rodrigues. Limitations of representation learning in small molecule property prediction. *Nature Communications*, 2023. doi:10.1038/s41467-023-41967-3.

[40] 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. doi:10.1021/ci00009a007.

[41] Urban Fagerholm, Sven Hellberg, Jonathan Alvarsson, Morgan Ekmefjord, and

Ola Spjuth. Comparing lipinski's rule of 5 and machine learning based prediction of fraction absorbed for assessing oral absorption in humans. *bioRxiv*, 2024. URL: https://www.biorxiv.org/content/early/2024/08/23/2024.08.20.608791, arXiv:https://www.biorxiv.org/content/early/2024/08/23/2024.08.20.608791.full.pdf, doi:10.1101/2024.08.20.608791.

[42] Wei Zhu, Yifan Wang, Yang Niu, Lei Zhang, and Zhen Liu. Current trends and challenges in drug-likeness prediction: Are they generalizable and interpretable? *Health Data Science*, 3:Article 0098, 2023. doi:10.34133/hds.0098.

[43] Bowen Li, Zhen Wang, Ziqi Liu, Yanxin Tao, Chulin Sha, Min He, and Xiaolin Li. DrugMetric: quantitative drug-likeness scoring based on chemical space distance. *Briefings in Bioinformatics*, 25(4):bbae321, 07 2024. arXiv:https://academic.oup.com/bib/article-pdf/25/4/bbae321/58468943/bbae321.pdf, doi:10.1093/bib/bbae321.

[44] Jinyu Sun, Ming Wen, Huabei Wang, Yuezhe Ruan, Qiong Yang, Xiao Kang, Hailiang Zhang, Zhimin Zhang, and Hongmei Lu. Prediction of drug-likeness using graph convolutional attention network. *Bioinformatics*, 38(23):5262–5269, 10 2022. arXiv:https://academic.oup.com/bioinformatics/article-pdf/38/23/5262/47465922/btac676.pdf, doi:10.1093/bioinformatics/btac676.

[45] E. J. Bjerrum. Smiles enumeration as data augmentation for neural network modeling of molecules, 2017. URL: https://arxiv.org/abs/1703.07076, arXiv:1703.07076.

[46] Benjamin Sanchez-Lengeling, Emily Reif, Adam Pearce, and Alexander B. Wiltschko. A gentle introduction to graph neural networks. *Distill*, 2021. https://distill.pub/2021/gnn-intro. doi:10.23915/distill.00033.

[47] Jianyuan Deng, Zhibo Yang, Hehe Wang, Iwao Ojima, Dimitris Samaras, and Fusheng Wang. Unraveling key elements underlying molecular property prediction: A systematic study, 2023. URL: https://arxiv.org/abs/2209.13492, arXiv:2209.13492.

[48] Simon Axelrod and Rafael Gomez-Bombarelli. Molecular machine learning with conformer ensembles, 2021. URL: https://arxiv.org/abs/2012.08452, arXiv:2012.08452.

[49] Jinsong Shao, Qineng Gong, Zeyu Yin, Wenjie Pan, Sanjeevi Pandiyan, and Li Wang. S2DV: converting SMILES to a drug vector for predicting the activity of anti-HBV small molecules. *Briefings in Bioinformatics*, 23(2):bbab593, 01 2022. arXiv:https://academic.oup.com/bib/article-pdf/23/2/bbab593/42805068/bbab593.pdf, doi:10.1093/bib/bbab593.

[50] S. Sadeghi, A. Bui, A. Forooghi, J. Lu, and A. Ngom. Can large language models understand molecules? *BMC Bioinformatics*, 25(1):225, 2024. URL: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-024-04567-8, doi:10.1186/s12859-024-04567-8.

[51] Yasuhiro Yoshikai, Tadahaya Mizuno, Shumpei Nemoto, and Hiroyuki Kusuhara. Difficulty in chirality recognition for transformer architectures learning chemical structures from string representations. *Nature Communications*, 15(1), February 2024. URL: http://dx.doi.org/10.1038/s41467-024-45102-8, doi:10.1038/s41467-024-45102-8.

[52] Katsuhisa Morita, Tadahaya Mizuno, and Hiroyuki Kusuhara. Investigation of a data split strategy involving the time axis in adverse event prediction using machine learning. *Journal of Chemical Information and Modeling*, 62(17):3982–3992, August 2022. URL: http://dx.doi.org/10.1021/acs.jcim.2c00765, doi:10.1021/acs.jcim.2c00765.

[53] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010. PMID: 20426451. arXiv:https://doi.org/10.1021/ci100050t, doi:10.1021/ci100050t.

[54] RDKit Community. *RDKit: Open-Source Cheminformatics Software*, 2023. Available at: https://rdkit.org/docs/Overview.html#what-is-it.

[55] Michael Dablander, Thierry Hanser, Renaud Lambiotte, et al. Exploring qsar models

for activity-cliff prediction. *Journal of Cheminformatics*, 15:47, 2023. doi:10.1186/s13321-023-00708-w.

[56] Ziqiao Zhang, Bangyi Zhao, Ailin Xie, Yatao Bian, and Shuigeng Zhou. Activity cliff prediction: Dataset and benchmark, 2023. URL: https://arxiv.org/abs/2302.07541, arXiv:2302.07541.

[57] Zachary A. Rollins, Amy C. Cheng, and Ehab Metwally. MolPROP: Molecular Property prediction with multimodal language and graph fusion. *Journal of Cheminformatics*, 16:56, 2024. doi:10.1186/s13321-024-00846-9.

[58] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation, 2019. URL: https://arxiv.org/abs/1902.07243, arXiv:1902.07243.

[59] Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. Qa-gnn: Reasoning with language models and knowledge graphs for question answering, 2022. URL: https://arxiv.org/abs/2104.06378, arXiv:2104.06378.

[60] Riccardo Smeriglio, Joana Rosell-Mirmi, Petia Radeva, and Jordi Abante. Leveraging protein-protein interactions in phenotype prediction through graph neural networks. In *2024 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–8, 2024. doi:10.1109/CIBCB58642.2024.10702170.

[61] Yuyang Wang, Zijie Li, and Amir Barati Farimani. *Graph Neural Networks for Molecules*, page 21–66. Springer International Publishing, 2023. URL: http://dx.doi.org/10.1007/978-3-031-37196-7_2, doi:10.1007/978-3-031-37196-7_2.

[62] M. Besharatifard and F. Vafaee. A review on graph neural networks for predicting synergistic drug combinations. *Artificial Intelligence Review*, 57:49, 2024. doi:10.1007/s10462-023-10669-z.

[63] Marco Proietti, Antonella Ragno, Beatrice L. Rosa, et al. Explainable ai in drug discovery: self-interpretable graph neural network for molecular property prediction using concept whitening. *Machine Learning*, 113:2013–2044, April 2024. doi:10.1007/s10994-023-06369-y.

[64] Wei Ju, Siyu Yi, Yifan Wang, Zhiping Xiao, Zhengyang Mao, Hourun Li, Yiyang Gu, Yifang Qin, Nan Yin, Senzhang Wang, Xinwang Liu, Xiao Luo, Philip S. Yu, and Ming Zhang. A survey of graph neural networks in real world: Imbalance, noise, privacy and ood challenges, 2024. URL: https://arxiv.org/abs/2403.04468, arXiv:2403.04468.

[65] Yushun Dong, Ninghao Liu, Brian Jalaian, and Jundong Li. Edits: Modeling and mitigating data bias for graph neural networks. In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 1259–1269, New York, NY, USA, 2022. Association for Computing Machinery. doi:10.1145/3485447.3512173.

[66] Hyunho Kim, Eunyoung Kim, Ingoo Lee, Bongsung Bae, Minsu Park, and Hojung Nam. Artificial intelligence in drug discovery: A comprehensive review of data-driven and machine learning approaches. *Biotechnology and Bioprocess Engineering*, 25:895–930, 12 2020. doi:10.1007/s12257-020-0049-y.

[67] Cheryl Lans and Tedje van Asseldonk. *Dr. Duke's Phytochemical and Ethnobotanical Databases, a Cornerstone in the Validation of Ethnoveterinary Medicinal Plants, as Demonstrated by Data on Pets in British Columbia*, pages 219–246. Springer International Publishing, Cham, 2020. doi:10.1007/978-3-030-44930-8_10.

[68] Adriano Rutz, Maria Sorokina, Jakub Galgonek, Daniel Mietchen, Egon Willighagen, Arnaud Gaudry, James G Graham, Ralf Stephan, Roderic Page, Jiří Vondrášek, Christoph Steinbeck, Guido F Pauli, Jean-Luc Wolfender, Jonathan Bisson, and Pierre-Marie Allard. The lotus initiative for open knowledge management in natural products research. *eLife*, 11:e70780, May 2022. doi:10.7554/eLife.70780.

[69] Qiujie Lv, Guanxing Chen, Haohuai He, Ziduo Yang, Lu Zhao, Kang Zhang, and Chen Chen. Tcmbank-the largest tcm database provides deep learning-based chinese-western

medicine exclusion prediction. *Signal Transduction and Targeted Therapy*, 8, 03 2023. doi:10.1038/s41392-023-01339-1.

[70] Zhaoqian Liu, Jie Dong, Hui Wei, Shao-Hua Shi, Ai-Ping Lu, Gui-Ming Deng, and Dong-Sheng Cao. Tcmsid: a simplified integrated database for drug discovery from traditional chinese medicine. *Journal of Cheminformatics*, 14, December 2022. doi:10.1186/s13321-022-00670-z.

[71] Sang-Kyun Kim, Myung-Ku Lee, Ho Jang, Jeong-Ju Lee, Sanghun Lee, Yunji Jang, Hyunchul Jang, and Anna Kim. Tm-mc 2.0: an enhanced chemical database of medicinal materials in northeast asian traditional medicine. *BMC Complementary Medicine and Therapies*, 24, January 2024. doi:10.1186/s12906-023-04331-y.

[72] Jia Ru, Peng Li, Jing Wang, Wei Zhou, Bo Li, Changtao Huang, Pengcheng Li, Zhihao Guo, Weixian Tao, Yin Yang, et al. Tcmsp: a database of systems pharmacology for drug discovery from herbal medicines. *Journal of Cheminformatics*, 6:1–6, 2014. doi:10.1186/1758-2946-6-13.

[73] Simon D Harding, Joanna L Sharman, Elena Faccenda, Christopher Southan, Adam J Pawson, Sean Ireland, Adam JG Gray, Lesley Bruce, Stephen P Alexander, Sally Anderton, et al. The iuphar/bps guide to pharmacology in 2020: extending immunopharmacology content and introducing the iuphar/mmv guide to malaria pharmacology. *Nucleic Acids Research*, 48(D1):D1006–D1021, 2020. doi:10.1093/nar/gkz951.

[74] Janna Hastings, Gareth Owen, Adriana Dekker, Marcus Ennis, Namrata Kale, Venkatesh Muthukrishnan, Steve Turner, Neil Swainston, Pedro Mendes, and Christoph Steinbeck. Chebi in 2016: Improved services and an expanding collection of metabolites. *Nucleic Acids Research*, 44(D1):D1214–D1219, 2016. doi:10.1093/nar/gkv1031.

[75] Anna Gaulton, Louisa J Bellis, A Patricia Bento, Jonathan Chambers, Matthew Davies, Anne Hersey, Yvonne Light, Sean McGlinchey, David Michalovich, Bissan Al-Lazikani, et al. Chembl: a large-scale bioactivity database for drug discovery. *Nucleic Acids Research*, 40(D1):D1100–D1107, 2012. doi:10.1093/nar/gkr777.

[76] Jeffrey A. van Santen, Grégoire Jacob, Amrit Leen Singh, Victor Aniebok, Marcy J. Balunas, Derek Bunsko, Fausto Carnevale Neto, Laia Castaño-Espriu, Chen Chang, Trevor N. Clark, Jessica L. Cleary Little, David A. Delgadillo, Pieter C. Dorrestein, Katherine R. Duncan, Joseph M. Egan, Melissa M. Galey, F.P. Jake Haeckl, Alex Hua, Alison H. Hughes, Dasha Iskakova, Aswad Khadilkar, Jung-Ho Lee, Sanghoon Lee, Nicole LeGrow, Dennis Y. Liu, Jocelyn M. Macho, Catherine S. McCaughey, Marnix H. Medema, Ram P. Neupane, Timothy J. O'Donnell, Jasmine S. Paula, Laura M. Sanchez, Anam F. Shaikh, Sylvia Soldatou, Barbara R. Terlouw, Tuan Anh Tran, Mercia Valentine, Justin J. J. van der Hooft, Duy A. Vo, Mingxun Wang, Darryl Wilson, Katherine E. Zink, and Roger G. Linington. The natural products atlas: An open access knowledge base for microbial natural products discovery. *ACS Central Science*, 5(11):1824–1833, 2019. PMID: 31807684. arXiv:https://doi.org/10.1021/acscentsci.9b00806, doi:10.1021/acscentsci.9b00806.

[77] Christopher M. Grulke, Antony J. Williams, Inthirany Thillanadarajah, and Ann M. Richard. Epa's dsstox database: History of development of a curated chemistry resource supporting computational toxicology research. *Computational Toxicology*, 12:100096, 2019. URL: https://www.sciencedirect.com/science/article/pii/S2468111319300234, doi:10.1016/j.comtox.2019.100096.

[78] Sunghwan Kim, Paul A Thiessen, Evan E Bolton, Jie Chen, Gang Fu, Andrius Gindulyte, Lianyi Han, Jie He, Sherry He, Benjamin A Shoemaker, et al. Pubchem 2019 update: improved access to chemical data. *Nucleic Acids Research*, 47(D1):D1102–D1109, 2019. doi:10.1093/nar/gky1033.

[79] David Wishart, Craig Knox, An Chi Guo, Savita Shrivastava, Murtaza Hassanali, Paul Stothard, Zhan Chang, and Jennifer Woolsey. Drugbank: A comprehensive resource for

in silico drug discovery and exploration. *Nucleic acids research*, 34:D668–72, 01 2006. doi:10.1093/nar/gkj067.

[80] A. Bari and Marina Gavrilova. Residual connection-based graph convolutional neural networks for gait recognition. *The Visual Computer*, 37, 09 2021. doi:10.1007/s00371-021-02245-9.

[81] Weimin Zhu, Yi Zhang, DuanCheng Zhao, Jianrong Xu, and Ling Wang. Hignn: Hierarchical informative graph neural networks for molecular property prediction equipped with feature-wise attention, 2022. URL: https://arxiv.org/abs/2208.13994, arXiv:2208.13994.

[82] Johannes Gasteiger, Florian Becker, and Stephan Günnemann. Gemnet: Universal directional graph neural networks for molecules, 2024. URL: https://arxiv.org/abs/2106.08903, arXiv:2106.08903.

[83] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL: http://arxiv.org/abs/1706.03762, arXiv:1706.03762.

[84] Ali Reza Sajun, Imran Zualkernan, and Donthi Sankalpa. A historical survey of advances in transformer architectures. *Applied Sciences*, 14(10), 2024. URL: https://www.mdpi.com/2076-3417/14/10/4316, doi:10.3390/app14104316.

[85] Ian T. Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016. URL: http://doi.org/10.1098/rsta.2015.0202, doi:10.1098/rsta.2015.0202.

[86] John M. Lee. *Introduction to Riemannian Manifolds*. Springer-Verlag, 2018.

[87] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2020. URL: https://arxiv.org/abs/1802.03426, arXiv:1802.03426.

[88] Hans-Peter Kriegel, Erich Schubert, and Arthur Zimek. The (black) art of runtime evaluation: Are we comparing algorithms or implementations? *Knowledge and Information Systems*, 52(2):341–378, 2016. doi:10.1007/s10115-016-1004-2.

[89] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 226–231. AAAI Press, 1996. URL: https://www.dbs.ifi.lmu.de/Publikationen/Papers/KDD-96.final.frame.pdf.

[90] Jian Pei, Vincent S. Tseng, Longbing Cao, Hiroshi Motoda, and Guandong Xu. Density-based clustering based on hierarchical density estimates. In Jian Pei, Vincent S. Tseng, Longbing Cao, Hiroshi Motoda, and Guandong Xu, editors, *Advances in Knowledge Discovery and Data Mining*, volume 7819 of *Lecture Notes in Computer Science*, chapter 14, pages 160–172. Springer, 2013. doi:10.1007/978-3-642-37456-2_14.

[91] Priya Poonia, Monika Sharma, Prakash Jha, and Madhu Chopra. Pharmacophore-based virtual screening of zinc database, molecular modeling and designing new derivatives as potential hdac6 inhibitors. *Molecular Diversity*, 27(5):2053–2071, 2023. doi:10.1007/s11030-022-10540-3.

[92] Markus Hofmarcher, Andreas Mayr, Elisabeth Rumetshofer, Peter Ruch, Philipp Renz, Johannes Schimunek, Philipp Seidl, Andreu Vall, Michael Widrich, Sepp Hochreiter, and Günter Klambauer. Large-scale ligand-based virtual screening for sars-cov-2 inhibitors using deep neural networks, 2020. URL: https://arxiv.org/abs/2004.00979, arXiv:2004.00979.

[93] Mahendra Awale, Xin Jin, and Jean-Louis Reymond. Stereoselective virtual screening of the zinc database using atom pair 3d-fingerprints. *Journal of Cheminformatics*, 7(1):3, 2015. doi:10.1186/s13321-014-0051-5.

[94] Neo Christopher Chung, BłaŻej Miasojedow, Michał Startek, and Anna Gambin. Jaccard/tanimoto similarity test and estimation methods for biological presence-absence data. *BMC Bioinformatics*, 20(S15), December 2019. URL: http://dx.doi.org/10.1186/s12859-019-3118-5, doi:10.1186/s12859-019-3118-5.

[95] DeepChem Developers. *DeepChem Documentation*, 2024. URL: https://deepchem.readthedocs.io/en/latest/.

[96] Python Software Foundation. *Python Standard Library: concurrent.futures*, 2024. URL: https://docs.python.org/3/library/concurrent.futures.html.

[97] Joseph B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7(1):48–50, 1956. URL: https://www.ams.org/journals/proc/1956-007-01/S0002-9939-1956-0078686-7/, doi:10.1090/S0002-9939-1956-0078686-7.

[98] SciPy Community. *SciPy: Open-source scientific computing tools for Python*, 2024. URL: https://scipy.org/.

[99] DGL Team. *Deep Graph Library (DGL) Documentation*, 2024. Accessed: 2024-11-24. URL: https://docs.dgl.ai/.

[100] PyTorch Geometric Team. *PyTorch Geometric Documentation*, 2024. URL: https://pyg.org/.

[101] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. URL: https://arxiv.org/abs/1607.06450, arXiv:1607.06450.

[102] Víctor Manuel Vargas, David Guijo-Rubio, Pedro Antonio Gutiérrez, and César Hervás-Martínez. Relu-based activations: Analysis and experimental study for deep learning. In Enrique Alba, Gabriel Luque, Francisco Chicano, Carlos Cotta, David Camacho, Manuel Ojeda-Aciego, Susana Montes, Alicia Troncoso, José Riquelme, and Rodrigo Gil-Merino, editors, *Advances in Artificial Intelligence*, pages 33–43, Cham, 2021. Springer International Publishing.

[103] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987. URL: https://www.sciencedirect.com/science/article/pii/0377042787901257, doi:10.1016/0377-0427(87)90125-7.

[104] Davoud Moulavi, Pablo Andretta Jaskowiak, Ricardo Campello, Arthur Zimek, and Joerg Sander. Density-based clustering validation. 04 2014. doi:10.1137/1.9781611973440.96.

[105] scikit-learn developers. *HDBSCAN — scikit-learn 1.5.2 documentation*, 2024. URL: https://scikit-learn.org/1.5/modules/generated/sklearn.cluster.HDBSCAN.html.

[106] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay. sklearn.cluster.kmeans, 2023. Accesso: 29 novembre 2024. URL: https://scikit-learn.org/1.5/modules/generated/sklearn.cluster.KMeans.html.

[107] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. *CoRR*, abs/1907.10902, 2019. URL: http://arxiv.org/abs/1907.10902, arXiv:1907.10902.

[108] T. Burt, K. S. Button, H. Thom, R. J. Noveck, and M. R. Munafò. The burden of the "false-negatives" in clinical development: Analyses of current and alternative scenarios and corrective measures. *Clinical and Translational Science*, 10(6):470–479, Nov 2017. Epub 2017 Jul 4. doi:10.1111/cts.12478.

[109] Shaohua Shi, Li Fu, Jiacai Yi, Ziyi Yang, Xiaochen Zhang, Youchao Deng, Wenxuan Wang, Chengkun Wu, Wentao Zhao, Tingjun Hou, Xiangxiang Zeng, Aiping Lyu,

and Dongsheng Cao. Chemfh: an integrated tool for screening frequent false positives in chemical biology and drug discovery. *Nucleic Acids Research*, 52(W1):W439–W449, 05 2024. arXiv:https://academic.oup.com/nar/article-pdf/52/W1/W439/58436182/gkae424.pdf, doi:10.1093/nar/gkae424.

[110] R. Sink, S. Gobec, S. Pecar, and A. Zega. False positives in the early stages of drug discovery. *Current Medicinal Chemistry*, 17(34):4231–4255, 2010. doi:10.2174/092986710793348545.

[111] Anja Struyf, Mia Hubert, and Peter Rousseeuw. Clustering in an object-oriented environment. *Journal of Statistical Software*, 1(4):1–30, 1997.

[112] Hui Zhao, Yuan Yang, Shuaiqi Wang, Xue Yang, Kaicheng Zhou, Caili Xu, Xuyao Zhang, Jiajun Fan, Dongyue Hou, Xingxiu Li, Hanbo Lin, Ying Tan, Shanshan Wang, Xin-Yi Chu, Dongzhi Zhuoma, Fengying Zhang, Dianwen Ju, Xian Zeng, and Yu Zong Chen. Npass database update 2023: quantitative natural product activity and species source database for biomedical research. *Nucleic Acids Research*, 51(D1):D621–D628, 11 2022. arXiv:https://academic.oup.com/nar/article-pdf/51/D1/D621/48441417/gkac1069.pdf, doi:10.1093/nar/gkac1069.

[113] National Library of Medicine. Medical Subject Headings (MeSH), 2024. Accessed: 2024-11-23. URL: https://www.nlm.nih.gov/mesh/meshhome.html.

[114] National Library of Medicine. PubMed, 2024. Accessed: 2024-11-23. URL: https://pubmed.ncbi.nlm.nih.gov/.

[115] Harry E. Pence and Antony Williams. Chemspider: An online chemical information resource. *Journal of Chemical Education*, 87(11):1123–1124, 2010. arXiv:https://doi.org/10.1021/ed100697w, doi:10.1021/ed100697w.

[116] Cymit Quimica. Cymit Quimica - Online Drug Marketplace, 2024. Accessed: 2024-11-23. URL: https://cymitquimica.com.

[117] Ambeed. Ambeed - Online Drug Marketplace, 2024. Accessed: 2024-11-23. URL: https://ambeed.com.

[118] BLD Pharm. BLD Pharm - Online Drug Marketplace, 2024. Accessed: 2024-11-23. URL: https://www.bldpharm.com.

[119] Biosynth. Biosynth - Online Drug Marketplace, 2024. Accessed: 2024-11-23. URL: https://www.biosynth.com.

[120] Elsevier. Elsevier - Journals, 2024. Accessed: 2024-11-23. URL: https://www.elsevier.com/products/journals?query=&page=1&sortBy=relevance.

[121] Japan Science and Technology Agency. J-global. https://jglobal.jst.go.jp/en/. Accessed: 2024-11-26.

[122] American Chemical Society. ACS Publications, 2024. Accessed: 2024-11-23. URL: https://pubs.acs.org/.

[123] Alireza Noruzi and Mohammadhiwa Abdekhoda. Google patents: The global patent search engine. *Webology*, 11, 06 2014.

[124] National Center for Biotechnology Information. Pubchem compound summary for cid 12304036, corycavamine. https://pubchem.ncbi.nlm.nih.gov/compound/12304036. Accessed: 2024-11-26.

[125] corycavamine [supplementary concept]. https://www.ncbi.nlm.nih.gov/mesh/2014144. Accessed: 2024-11-26.

[126] Cymit Química S.L. Corycavine. https://cymitquimica.com/products/3D-FC65473/521-85-7/corycavine/. Accessed: 2024-11-26.

[127] National Center for Biotechnology Information. Pubchem compound summary for cid 72322, coptisine. https://pubchem.ncbi.nlm.nih.gov/compound/72322. Accessed: 2024-11-26.

[128] Dan Yu, Bang-Bao Tao, Yun-Yun Yang, Li-Sha Du, Shuang-Shuang Yang, Xiao-Jie He, Yu-Wen Zhu, Jun-Kai Yan, and Qing Yang. The ido inhibitor coptisine ameliorates

cognitive impairment in a mouse model of alzheimer's disease. *Journal of Alzheimer's Disease*, 43(1):291–302, 2015. Accessed: 2024-11-26. URL: https://pubmed.ncbi.nlm.nih.gov/25079795, doi:10.3233/JAD-140414.

[129] Korea Institute of Oriental Medicine. Tm-mc 2.0: an enhanced chemical database of medicinal materials in northeast asian traditional medicine. https://tm-mc.kr/detail.do. Accessed: 2024-11-26.

[130] National Center for Biotechnology Information. Pubchem compound summary for cid 441934, notoginsenoside r1. https://pubchem.ncbi.nlm.nih.gov/compound/441934. Accessed: 2024-11-26.

[131] European Bioinformatics Institute. Notoginsenoside r1 (chebi:77149). https://www.ebi.ac.uk/chebi/searchId.do?chebiId=CHEBI:77149. Accessed: 2024-11-26.

[132] European Bioinformatics Institute. Compound report card: Notoginsenoside r1 (chembl507115). https://www.ebi.ac.uk/chembl/compound_report_card/CHEMBL507115/. Accessed: 2024-11-26.

[133] Philip Lam, Flora Cheung, Hiu-Yee Tan, Ningsan Wang, Man-Fung Yuen, and Yibin Feng. Hepatoprotective effects of chinese medicinal herbs: A focus on anti-inflammatory and anti-oxidative activities. *International Journal of Molecular Sciences*, 17(4):465, Mar 2016. doi:10.3390/ijms17040465.

[134] National Center for Biotechnology Information. Pubchem compound summary for cid 100952311, 2-[3,4-dihydroxy-5-[(2s,3r,4s,5r)-3,4,5-trihydroxyoxan-2-yl]oxyphenyl]-3,5,7-trihydroxychromen-4-one. https://pubchem.ncbi.nlm.nih.gov/compound/100952311. Accessed: 2024-11-26.

[135] Japan Science and Technology Agency. Myricetin 3'-xyloside. https://jglobal.jst.go.jp/en/detail?JGLOBAL_ID=200907075810902140. Accessed: 2024-11-26.

[136] Irina O. Vvedenskaya, Robert T. Rosen, Joseph E. Guido, David J. Russell, Kenneth A. Mills, and Nicholi Vorsa. Characterization of flavonols in cranberry (*Vaccinium macrocarpon*) powder. *Journal of Agricultural and Food Chemistry*, 52(2):188–195, Jan 2004. doi:10.1021/jf034970s.

[137] National Center for Biotechnology Information. Pubchem compound summary for cid 5280343, quercetin. https://pubchem.ncbi.nlm.nih.gov/compound/5280343. Accessed: 2024-11-26.

[138] U.S. Environmental Protection Agency. Chemical data reporting under the toxic substances control act. https://www.epa.gov/chemical-data-reporting. Accessed: 2024-11-26.

[139] National Center for Biotechnology Information. Pubchem compound summary for cid 7795, isoamyl butyrate. https://pubchem.ncbi.nlm.nih.gov/compound/Isoamyl-butyrate. Accessed: 2024-11-26.

[140] National Center for Biotechnology Information. Pubchem compound summary for cid 5352539, cis-3-hexenyl isobutyrate. https://pubchem.ncbi.nlm.nih.gov/compound/cis-3-Hexenyl-isobutyrate. Accessed: 2024-11-26.