2024

Advanced Mathematical Graph-based Machine Learning and Deep Learning Models for Drug Design

Farjana Tasnim Mukta
University of Kentucky, farjanamasud143@gmail.com
Author ORCID Identifier:
https://orcid.org/0009-0006-1151-4420
Digital Object Identifier: https://doi.org/10.13023/etd.2024.373

STUDENT AGREEMENT:

I represent that my thesis or dissertation and abstract are my original work. Proper attribution has been given to all outside sources. I understand that I am solely responsible for obtaining any needed copyright permissions. I have obtained needed written permission statement(s) from the owner(s) of each third-party copyrighted matter to be included in my work, allowing electronic distribution (if such use is not permitted by the fair use doctrine) which will be submitted to UKnowledge as Additional File.

I hereby grant to The University of Kentucky and its agents the irrevocable, non-exclusive, and royalty-free license to archive and make accessible my work in whole or in part in all forms of media, now or hereafter known. I agree that the document mentioned above may be made available immediately for worldwide access unless an embargo applies.

I retain all other ownership rights to the copyright of my work. I also retain the right to use in future works (such as articles or books) all or part of my work. I understand that I am free to register the copyright to my work.

REVIEW, APPROVAL AND ACCEPTANCE

The document mentioned above has been reviewed and accepted by the student's advisor, on behalf of the advisory committee, and by the Director of Graduate Studies (DGS), on behalf of the program; we verify that this is the final, approved version of the student's thesis including all changes required by the advisory committee. The undersigned agree to abide by the statements above.

Farjana Tasnim Mukta, Student

Dr. Duc Nguyen, Major Professor

Dr. Benjamin Braun, Director of Graduate Studies

Advanced Mathematical Graph-based Machine Learning and Deep Learning Models for Drug Design

DISSERTATION

A dissertation submitted in partial
fulfillment of the requirements for
the degree of Doctor of Philosophy
in the College of Arts and Sciences
at the University of Kentucky

By
Farjana Tasnim Mukta
Lexington, Kentucky

Director: Dr. Duc Nguyen, Professor of Mathematics
Lexington, Kentucky
2024

https://orcid.org/0009-0006-1151-4420

ABSTRACT OF DISSERTATION

Advanced Mathematical Graph-based Machine Learning and Deep Learning Models for Drug Design

Drug discovery is a highly complicated and time-consuming process. One of the main challenges in drug development is predicting whether a drug-like molecule will interact with a specific target protein. This prediction accelerates target validation and drug development. Recent research in biomolecular sciences has shown significant interest in algebraic graph-based models for representing molecular complexes and predicting drug-target binding affinity. In this thesis, we present algebraic graph-based molecular representations to create data-driven scoring functions (SF) using extended atom types to capture wide-range interactions between targets and drug candidates. Our model employs multiscale weighted colored subgraphs for the protein-ligand complex, colored based on SYBYL atom types. Utilizing machine learning and deep learning techniques such as gradient-boosting decision trees (GBDT), random forests (RF), support vector machines (SVM), extreme gradient boosting (XGBoost), convolutional neural networks (CNNs), and graph convolutional neural networks (GCNs), our SF outperformed numerous state-of-the-art models in various PDBbind benchmark datasets for binding affinity scoring power, the D3R dataset, a worldwide grand challenge in drug design, and various blood-brain permeability prediction datasets.

KEYWORDS: Machine Learning, Protein-Ligand Binding Affinity Prediction, Weighted Colored Subgraph, Blood-Brain Permeability Prediction

Farjana Tasnim Mukta

August 8, 2024

Advanced Mathematical Graph-based Machine Learning and Deep Learning Models for Drug Design

By
Farjana Tasnim Mukta

Dr. Duc Nguyen
Director of Dissertation

Dr. Benjamin Braun
Director of Graduate Studies

August 8, 2024
Date

Dedicated to my Mom and Dad

ACKNOWLEDGMENTS

First and foremost, I would like to extend my deepest gratitude to my advisor, Dr. Duc Nguyen, for their unwavering support, guidance, and encouragement throughout my research. Your expertise and insights have been invaluable, and your patience and dedication have continually inspired me to strive for excellence.

I am also profoundly thankful to my committee members, Dr. Qiang Ye, Dr. David Murrugarra, Dr. Hunter Moseley, and Dr. Sally Ellingson for their constructive feedback and invaluable advice. Your critical comments and suggestions have significantly enriched this work.

A special thanks to my colleagues and friends in the Department of Mathematics at the University of Kentucky. Your friendship and support have made this journey both enjoyable and fulfilling.

On a personal note, I am deeply indebted to my family for their endless love, patience, and encouragement. To my parents, sisters, and brothers, thank you for believing in me and for your unwavering support. To my beloved husband, Masud Rana, your understanding, encouragement, and continuous support have been my source of strength. Each day, you inspire me with your wisdom and kindness, making our shared dreams possible. Thank you for walking every step of this journey by my side, especially during those moments when the path seemed impossible. To my son, Aryaav Raanan Masud, you are my greatest joy and motivation; your smiles and laughter have brightened even the toughest days.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1 Drug Discovery Introduction

1.1 Process of Drug Discovery

Developing new drugs is both time-consuming and expensive, encompassing the entire journey from research and development (R&D) to market approval. Navigating a drug through testing and trials to clinical application involves several phases, each with its own potential for failure, which can result in a significant amount of costs, especially in the later stages [31]. Identifying or designing truly novel drugs is scientifically challenging and commercially risky, as it requires substantial investment and carries a high degree of uncertainty. This complex, interdisciplinary process integrates knowledge and techniques from various scientific fields, including structural biology, pharmacology, and computer science, to identify new candidate medications [34]. The integration of these diverse disciplines is crucial for addressing the multifaceted challenges of drug discovery. For instance, biological studies, such as genomics, transcriptomics, and proteomics, help in understanding disease mechanisms and identifying potential drug targets by analyzing gene expression and protein interactions [38]. Chemistry is essential for designing and synthesizing new compounds, employing techniques like medicinal chemistry and combinatorial chemistry to create molecules with desired biological activities [114]. Pharmacology provides insights into the drug's effects on biological systems through pharmacokinetic and pharmacodynamic studies, which help in understanding how drugs are absorbed, distributed, metabolized, and excreted in the body [104]. Computational science offers powerful tools for modeling, simulating, and analyzing data throughout the drug discovery process, using methods such as molecular docking, quantitative structure-activity relationship (QSAR) models, and machine learning algorithms to predict the behavior of potential drug candidates [35].

The modern drug discovery process has evolved over decades and now includes several key phases: target identification and validation, hit generation, lead optimization, preclinical development, and clinical trials, which typically takes 10 to 15 years for the whole process [31, 99], as presented in Figure 1.1. Each of these stages employs computational methods to enhance efficiency and precision, especially after the drug target is identified. The following sections will delve into each phase of the drug discovery process, with a particular focus on the role of computational techniques.

Target Identification and Validation The target identification and validation phase in drug discovery involves identifying a biological target linked to a specific disease. These targets are usually proteins, such as enzymes, receptors, or ion channels, that are integral to the disease's pathology. Research has shown that a significant portion of drug development failures stem from incorrect target selection [106]. This underscores the importance of confirming that small molecules designed to bind a particular target will effectively treat or mitigate the disease in patients. Once a target is

identified, its role in the disease is validated through various biological experiments. This implies that modulating the target with a drug will have a therapeutic effect.

In this phase, computational methods are extensively used alongside bioinformatics tools to analyze complex biological datasets. Innovations in technology have significantly enhanced the capacity to process and interpret large volumes of genomic, transcriptomic, proteomic, and metabolomic data, allowing researchers to identify potential targets with greater precision. For example, machine learning algorithms can detect patterns in extensive datasets that may suggest a protein's involvement in disease pathology. Additionally, systems biology approaches enable the modeling of intricate biological networks, which aids in predicting the impact of targeting a specific protein within the overall system [35, 38].

Once potential targets are identified, their roles in the disease are validated through rigorous biological experiments, such as gene knockout studies, RNA interference, and CRISPR-Cas9 gene editing [42, 33, 110]. This thorough validation process supports that modulating the target with a drug will produce a therapeutic effect, thereby increasing the likelihood of successful treatment outcomes.

Identifying small molecule inhibitors or activators remains a fundamental aspect of drug discovery. Therefore, this thesis will concentrate on the computational methods and tools developed for discovering small molecule drug candidates. These approaches are pivotal in enhancing the efficiency and precision of the drug discovery process, ultimately aiding in the identification of promising therapeutic compounds.

Hit Generation After identifying and validating a drug target, the drug discovery process moves to the hit identification stage, which focuses on discovering 'hit' compounds that demonstrate strong binding affinity, typically in the micromolar range or better, for the target. Binding affinity is the primary criterion at this stage; however, it is equally important to evaluate other pharmacokinetic properties such as ADMET (absorption, distribution, metabolism, excretion, and toxicity). These properties are crucial in determining whether a compound is a viable drug candidate, ensuring it not only binds effectively to the target but also possesses the necessary characteristics for further development.

After analyzing the properties of a long list of approved drugs, researchers employ various computational and experimental techniques to identify hit compounds. High-throughput screening (HTS) is a widely used method that allows the rapid testing of thousands to millions of compounds for activity against the target in order to identify the 'hit'. HTS involves the use of robotics, automated workflows, and sensitive detection methods to identify compounds that exhibit desired biological activity [82].

Computational methods are used to screen compound libraries in silico, predicting which compounds are likely to bind to the target based on molecular modeling and docking simulations. Techniques such as molecular docking, quantitative structure-activity relationship (QSAR) models, and machine learning algorithms are employed to identify promising candidates from large datasets [107].

The hit identification stage is critical because it lays the foundation for subsequent lead optimization and preclinical development. Identified hits undergo further valida-

tion and optimization to enhance their binding affinity, selectivity, and pharmacokinetic properties. This iterative process ensures that the most promising compounds advance to the next stages of drug development, ultimately increasing the likelihood of successful drug candidates.

Lead Optimization After the hit identification stage, the drug discovery process advances to lead optimization. This phase aims to refine the identified hits into leads with improved binding affinity, efficacy, selectivity, and pharmacokinetic properties, ensuring they meet the stringent criteria for a viable drug candidate.

The initial step in lead optimization involves confirming the activity of the hits identified during the screening phase. This is achieved through secondary assays that validate the primary screening results, ensuring reproducibility and accuracy [82]. One of the core aspects of lead optimization is the study of Structure-Activity Relationships (SAR). This involves systematic modifications to the chemical structure of the hits to identify changes that enhance biological activity and reduce off-target effects. Computational chemistry techniques, such as molecular dynamics simulations and energy minimization, are often employed to predict the effects of these modifications and guide the design of new analogs [107].

In addition to optimizing binding affinity and selectivity, lead optimization also focuses on improving pharmacokinetic properties, including absorption, distribution, metabolism, and excretion (ADME). These properties are critical for ensuring that the drug reaches the target site in the body at therapeutic concentrations without causing toxicity. High-throughput ADME screening assays and in silico models, such as physiologically based pharmacokinetic (PBPK) models, are used to predict and optimize these properties [63]. Ensuring that the lead compounds are safe and non-toxic is another critical aspect of lead optimization. Comprehensive in vitro and in vivo toxicology studies are conducted to evaluate potential adverse effects. Computational toxicology models can predict toxicological outcomes based on the chemical structure, helping to identify and mitigate potential safety issues early in the development process.

Enhancing the selectivity of lead compounds to ensure they specifically interact with the target protein without affecting other biological pathways is essential, as off-target effects can lead to undesirable side effects and toxicity. High-throughput screening against panels of related proteins, along with in silico modeling, helps identify and mitigate these off-target interactions [13]. Lead optimization is an iterative process involving cycles of design, synthesis, and testing. Each iteration aims to refine the properties of the lead compounds, guided by data from biological assays and computational predictions. This approach ensures that the lead compounds progressively meet the desired criteria for efficacy, safety, and pharmacokinetics. Overall, lead optimization is a critical stage in drug discovery that transforms promising hits into potential drug candidates by improving their pharmacological profiles and ensuring they are suitable for further development.

Preclinical Development The preclinical development stage is a critical phase in drug discovery that involves comprehensive testing of lead compounds to ensure their safety and efficacy before proceeding to clinical trials in humans. This stage encompasses several key activities, including toxicology studies, pharmacokinetics (PK) and pharmacodynamics (PD) evaluations, and in vivo efficacy testing.

One of the primary goals of preclinical development is to assess the potential toxicity of the lead compounds. Toxicology studies are conducted in vitro (e.g., using cell cultures) and in vivo (e.g., using animal models) to evaluate the safety profile of the compound. These studies help identify any adverse effects and establish safe dosage ranges. Regulatory guidelines require extensive toxicological evaluation to ensure that the compound does not pose significant risks to human health.

In vivo efficacy testing is conducted in this phase to demonstrate the compound's therapeutic potential in animal models of the disease. These studies aim to confirm that the compound can effectively modulate the target and produce the desired therapeutic effects, providing valuable insights into efficacy, pharmacodynamics, and potential side effects.

Preclinical development also involves identifying and mitigating potential risks, such as off-target effects, drug-drug interactions, and long-term toxicity. Computational toxicology models and high-throughput screening assays help predict and address these risks early in the development process [78].

Clinical Trials The clinical trial stage is a crucial phase in drug development, involving human participants to evaluate the safety, efficacy, and optimal dosing of the drug. Given the high costs associated with clinical trials, only the most promising compounds progress to this stage. Compounds exhibiting undesirable properties, such as off-target effects, high toxicity, or low solubility, are usually excluded from further development. As per the FDA website [43], this stage is divided into four phases: Phase I, Phase II, Phase III, and Phase IV, each with distinct objectives and methodologies. Phase I trials are the first stage of testing in human subjects and are primarily focused on assessing the safety, tolerability, pharmacokinetics (PK), and pharmacodynamics (PD) of the drug. Typically, these studies involve less than a hundred healthy volunteers or patients and aim to determine the optimal dosage to maximize therapeutic benefits without causing intolerable side effects. Approximately 70% of drug candidates successfully move on to Phase II. In Phase II trials, the primary goal is to evaluate the drug's efficacy while continuing to assess its safety. This phase involves up to several hundred patients who have the condition that the drug is intended to treat. Researchers gather additional safety data and refine the dosing regimen based on the results. Phase II trials help establish the therapeutic potential of the drug and design protocols for the next phase. About 33% of drugs in Phase II progress to Phase III. Phase III trials are large-scale studies involving several hundred to several thousand patients to confirm the drug's efficacy, monitor side effects, and compare it to existing treatments. Successful Phase III trials are essential for submitting a New Drug Application (NDA) to regulatory agencies such as the FDA or EMA. Phase IV, or post-marketing surveillance, occurs after the drug

has been approved and marketed. This phase monitors the drug for any adverse effects or long-term efficacy issues in the general population.

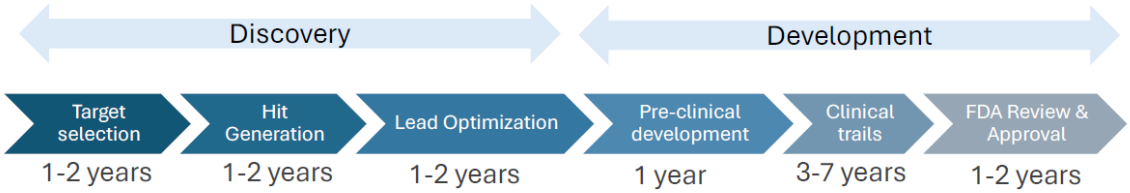

Figure 1.1: Drug Discovery Pipeline

1.2 Application of Machine Learning in Drug Discovery

Machine learning (ML) and its subset, deep learning, have revolutionized the field of drug discovery by providing advanced methods to analyze vast amounts of data, predict drug-target interactions, optimize drug properties, and identify potential drug candidates. In recent years, the integration of machine learning (ML) and artificial intelligence (AI) into drug discovery has created new opportunities for innovation [124, 47]. These technologies significantly accelerate the drug discovery process and reduce costs by learning from historical data and making accurate predictions. This section explores the various applications of machine learning in drug discovery, highlighting key techniques and their impact on the industry.

A crucial aspect of in silico drug development is predicting drug-protein interactions using diverse biological data sources. This task is complicated by the vast number of unknown interactions, which can hinder the accuracy of large-scale predictions. For instance, Wei, Zhang, and Gong (2022) [133] introduced DeepLPI, a deep learning model aimed at predicting protein-ligand interactions to facilitate drug repurposing. The model integrates ResNet-based 1-dimensional convolutional neural networks (1D CNNs) and bi-directional long short-term memory networks (biLSTMs) to process raw molecular sequences and protein sequences. DeepLPI was trained and evaluated on datasets from BindingDB and Davis, and also tested on a COVID-19 dataset. A decision tree-based meta-classifier had been developed by Costa et al. (2010) [23] to predict morbid and druggable human genes on a genome-wide scale, where "morbid" refers to genes associated with diseases whose mutations or variations can lead to pathological conditions. The classifier was trained using network topology features of protein-protein, metabolic, and transcriptional interactions, along with tissue expression and subcellular localization data. Their approach correctly identified 66% of known morbid genes and 78% of known druggable genes. This work demonstrates the potential of using systems-level data and machine learning to accelerate the discovery of gene-disease associations and druggable targets. Whereas, Jeon et al. (2014)

[60] developed a support vector machine (SVM)-based model to classify proteins as drug targets or non-drug targets specifically for breast, pancreatic, and ovarian cancers. This SVM classifier was trained using various genomic datasets, including gene expression profiles, sequence information, and structural data. By integrating these diverse data sources, the model could effectively capture the complex relationships and features that distinguish drug targets from non-drug targets.

Virtual screening (VS) is another significant application where ML models evaluate vast libraries of compounds to identify those most likely to bind to a specific target. By predicting the binding affinity of compounds, ML significantly reduces the number of compounds that need to be tested experimentally. VS for hit and lead identification typically involves two main approaches: structure-based (SBVS) and ligand-based (LBVS) [70]. In SBVS, known protein structures are used to dock small molecules from a compound library into the target's active site. A mathematical scoring function evaluates binding tightness, and a post-processing step ranks compounds based on these scores. The top-ranked compounds are selected for further experimental testing. Various docking programs with different sampling algorithms and scoring functions are available for this purpose, such as AUTODOCK [88], Autodock Vina [123], DOCK [40], and GOLD [62]. On the other hand, LBVS relies on the properties of known active ligands to predict the activity of new compounds. This approach uses chemical and physical similarities between compounds to identify potential drug candidates, without requiring the 3D structure of the target protein. Techniques such as quantitative structure-activity relationships (QSAR) and similarity searches [77, 83] are commonly employed in LBVS.

In addition to structure-based computational methods, such as protein-ligand binding affinity scoring functions and molecular docking, machine learning-based techniques like de novo fragment growth methods [57] have also significantly advanced the field of drug discovery. ML based de novo fragment growth methods [29, 50], generate new molecular structures from scratch, focusing on growing fragments of molecules to optimize their properties. This technique has proven to be effective in exploring chemical spaces that were previously inaccessible, leading to the discovery of novel compounds with desirable pharmacological characteristics. In many ways, these methods can be viewed as a form of in silico combinatorial chemistry, where large virtual libraries of molecular structures are generated and evaluated for potential biological activity. This approach leverages computational power to efficiently explore vast chemical spaces and identify promising candidates for further development.

1.3 Machine Learning Algorithms

Machine learning (ML) and deep learning, a subset of ML, have become an essential tool in drug discovery, providing powerful methods to analyze vast amounts of data, predict drug-target interactions, optimize drug properties, and identify potential drug candidates. These algorithms can learn from historical data and make accurate predictions, significantly accelerating the drug discovery process and reducing costs. Machine learning techniques can be broadly classified into several categories based

on the nature of the training data, the manner in which this data is presented to the model, and the criteria used for evaluating the algorithm's performance [87].

- Supervised Learning: In supervised learning, the model is trained using a dataset of labeled examples and then generates predictions for new, unseen data. This approach is highly effective for tasks such as predicting drug-target interactions and optimizing drug properties. Techniques like support vector machines (SVMs), decision trees, and deep neural networks are commonly used in supervised learning. These methods are widely applied in classification, regression, and ranking problems, making them valuable for virtual screening and QSAR modeling [12].

- Unsupervised Learning: In unsupervised learning, the model is provided exclusively with unlabeled training data and is responsible for identifying patterns and making predictions based on this data. Without labeled examples, quantitatively evaluating the model's performance can be challenging. However, unsupervised learning is particularly valuable for identifying inherent patterns and structures within the data. In graph learning, for example, the model processes unlabeled data to uncover patterns, with node clustering being a common unsupervised learning task. This approach is especially useful for tasks such as clustering similar compounds and discovering hidden relationships in chemical datasets [51].

- Semi-supervised Learning: Semi-supervised learning uses a combination of labeled and unlabeled data to train models. By utilizing the large amounts of available unlabeled data, this approach improves predictive accuracy and reduces dependence on labeled data for a variety of problems from classification and regression to clustering and association. This method is particularly useful when acquiring labeled data is expensive or time-consuming. Common techniques in this category include self-training, co-training, and graph-based methods [20].

1.3.1 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a nonprobabilistic kernel-based supervised learning method that operates by mapping input vectors into a high-dimensional feature space where a decision hyperplane is constructed to delineate between different classification or regression targets [125]. The main objective of the SVM algorithm is to find the optimal hyperplane in an N-dimensional space that can effectively separate the data points belonging to different classes within the feature space. The decision hyperplane in SVM is defined by the equation:

$$\omega \cdot x + b = 0 \tag{1.1}$$

where ω represents the weight vector, x denotes the input feature vector, and b the bias term. The hyperplane aims to maximize the margin between the closest points

of different classes, known as support vectors. The margin is the distance between the hyperplane and the nearest data points from each class, which forms the positive and negative hyperplanes, see Figure 1.2. The hyperplane aims to maximize the margin between the closest points of different classes, ensuring robust classification. The dimension of the hyperplane is determined by the number of features present in the dataset, as it spans across the feature space to delineate between classes. The hyperplane is defined by a decision function

$$f(x) = \omega \mathbf{x} + \mathbf{b} \tag{1.2}$$

where ω represents the weight vector, $\mathbf{x}$ denotes the input feature vector, and $\mathbf{b}$ denotes the bias term.

During SVM training, the goal is to minimize the combination of a regularization term and an error term, while ensuring that errors are constrained to a specified tolerance level. The regularization term, often denoted by the parameter C governs the trade-off between achieving a broad margin and minimizing errors. An excessively large value of C might induce overfitting by compelling the model to incorporate unnecessary support vectors to avoid the high penalty for non-separable points. Conversely, choosing a very small value of C could lead to underfitting, as the model might not capture the complexities the data adequately. SVM utilizes kernel functions like linear, polynomial, and radial basis function (RBF) to transform the input feature space into higher dimensions, allowing for the representation of nonlinear relationships. Parameter γ in the RBF kernel controls the width of the kernel and, consequently, influences the generalization capability of the SVM model. A large γ value might result in overfitting, as the support vectors would only have a substantial influence on a small region nearby. Conversely, a small γ value may lead to underfitting by restricting the model's capacity to incorporate information from the input data. The fine-tuning of parameters like C and γ plays an important role in shaping its effectiveness across diverse datasets.

1.3.2 Random Forest (RF)

The Random Forest algorithm is a robust tree-learning technique widely utilized in machine learning [16], which has seen significant application in predicting binding affinity problems [9, 130]. It offers several advantages, notably its ability to handle feature selection without complex procedures. RF is known for its robustness to parameter settings and its resilience in the face of redundant features.

RF constructs multiple decision trees during training and aggregates their outputs to improve accuracy and control overfitting. The process begins with bootstrapping, where each tree is trained on a different subset of the training data. These subsets are created by randomly sampling the training data with replacement, ensuring that each tree receives a slightly different dataset.

Each tree in the forest is a decision tree that splits the data based on feature values to make predictions. These splits are determined by metrics such as information gain for classification or variance reduction for regression. Unlike traditional decision trees that consider all possible features for splits, Random Forests introduce randomness

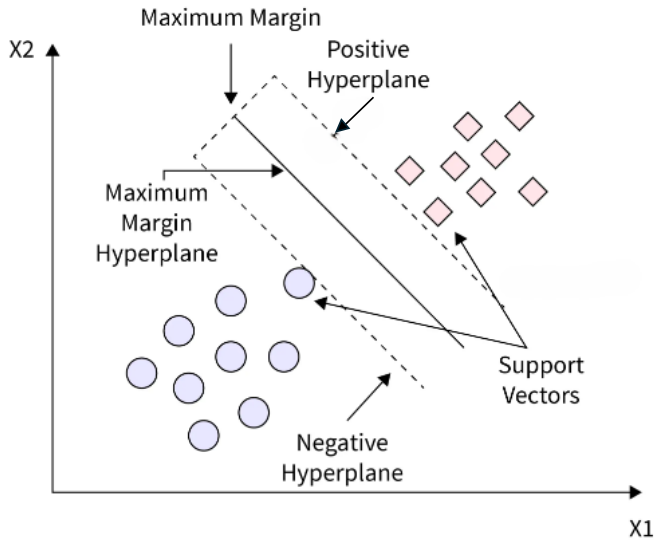

Figure 1.2: A diagram of Support Vector Machine algorithm [Image: Gupta, 2023]

by allowing each node to consider only a random subset of features. This approach enhances diversity among the trees and reduces their correlation, making the ensemble more robust.

The trees in a Random Forest are grown to their maximum depth without pruning, which means each tree is fully grown and can potentially overfit the training data. However, the ensemble of many overfitted trees reduces the overall variance, leading to a more robust model.

For classification tasks, each tree in the forest votes for a class, and the class with the majority vote is chosen as the final prediction, see Figure 1.3. For regression tasks, each tree predicts a numerical value, and the final prediction is the average of these values.

Several hyperparameters can be tuned to optimize the performance of a Random Forest model. The number of trees (`n_estimators`) is a critical parameter; increasing the number usually improves performance but also increases computational cost. The default value for `n_estimators` is 100. The maximum depth (`max_depth`) of each tree can be controlled to prevent overfitting, with the default value being `None`, which means nodes are expanded until all leaves are pure or contain fewer than `min_samples_split` samples. Parameters like the minimum samples required to split an internal node (`min_samples_split`, default value of 2) and the minimum samples required to be at a leaf node (`min_samples_leaf`, default value of 1) can also be adjusted to smooth the model and control its complexity. The `bootstrap` parameter (default value `True`) determines whether bootstrap samples are used when building trees, and the `criterion` parameter specifies the function used to measure the quality of a split. For classification, common criteria include "gini" (Gini impurity, default) and "entropy" (information gain), while for regression, options include "mse" (mean

squared error, default) and "mae" (mean absolute error).

Mathematically, *Gini impurity* for classification tasks is a measure of how often a randomly chosen element would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset. It is defined as:

$$\text{Gini}(D) = 1 - \sum_{k=1}^{K} p_k^2$$

where p_k is the proportion of samples belonging to class k in dataset D.

For regression tasks, the *Mean Squared Error (MSE)* measures the quality of a split by assessing the average squared difference between the observed actual outcomes and the outcomes predicted by the model. It is defined as:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

where y_i are the true values, $\hat{y}_i$ are the predicted values, and N is the number of samples.

In regression tasks, the ensemble prediction in Random Forest is derived by averaging the predictions of individual decision trees. Conversely, for classification tasks, the ensemble prediction involves voting among the predictions of the trees. This versatile approach allows Random Forest to effectively handle high-dimensional data, capture complex relationships, and mitigate overfitting, making it a widely used technique in machine learning.

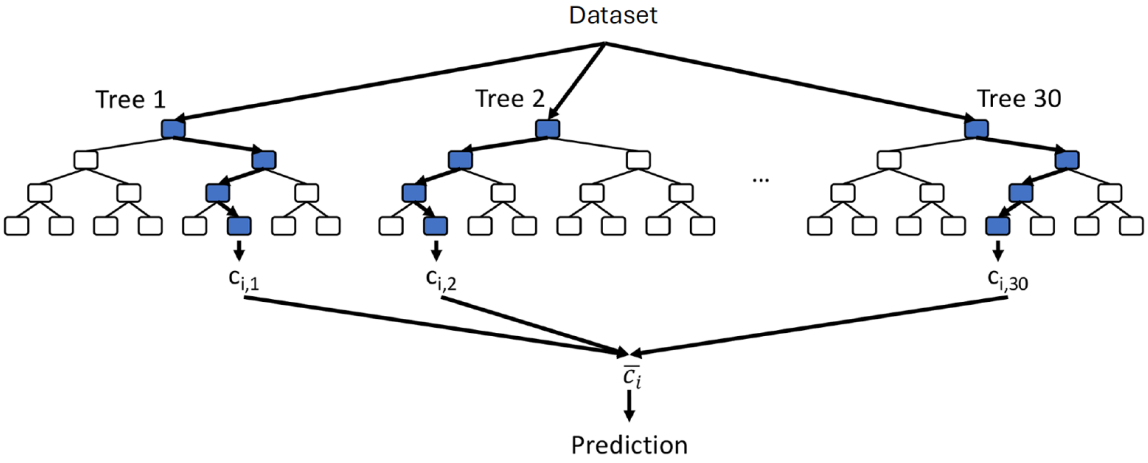

Figure 1.3: A diagram of random forest algorithm [Image: Evans et al., 2019]

1.3.3 Gradient Boosting Decision Trees (GBDT)

Gradient Boosting Decision Trees (GBDT) is a potent supervised learning technique that merges gradient descent optimization with decision trees [44], effectively combining the strengths of both approaches to create a robust and versatile ensemble learning algorithm, where each decision tree acts as a weak learner contributing to the collective predictive power of the model. It operates on a dataset $\{x^{(i)}, y^{(i)}\}_{i=1}^{M}$, consisting of input features $x^{(i)}$ and their corresponding target values $y^{(i)}$, aiming to construct an ensemble of T decision trees. Every tree is trained to minimize a specific loss function $\mathcal{L}(y_i, f(x_i))$ given by

$$\arg\min_{f} \sum_{i} l_i(y_i, f(x_i))$$

In this approach, individual decision trees are trained sequentially and are assembled in a stagewise manner to enhance their ability to learn complex features. Each subsequent tree corrects the residuals of the previous iteration. The ensemble prediction is computed by aggregating the predictions of all trees as follows,

$$\hat{y}_i = f(x_i) = \sum_{t=1}^{T} \nu f_t(x; \Theta_t)$$

where $\{f_t(x; \Theta_t)\}_{t=1}^{T}$ are incremental functions with initial value $\hat{y}^{(0)} = 0$ and ν is the learning rate. During each iteration, the negative gradient of the loss function

$$r_i^{(n)} = -\left[\frac{\partial \mathcal{L}(y^{(i)}, f(x_i))}{\partial f(x_i)}\right]_{f(x)=\hat{y}^{(t-1)}}$$

is computed with respect to the previous iteration's prediction. This gradient represents the residual errors that need to be corrected by the new decision tree. The next step involves finding the new tree $f_t(x; \Theta_t)$ that minimizes the sum of squared differences between these residuals and the tree predictions::

$$f_t = \arg\min_{h} \sum_{i=1}^{M} (r_i^{(n)} - f(x_i))^2$$

After that a optimal weight γ_t for the new tree $f_t(x; \Theta_t)$ has been computed

$$\gamma_t = \arg\min_{\gamma} \sum_{i=1}^{M} \mathcal{L}(y_i, \hat{y}^{t-1}(x_i) + \nu f_t(x_i))$$

After computing the optimal weight γ_t for the new tree $f_t(x; \Theta_t)$, the model is updated by incorporating the weighted contribution of this new tree into the existing model. This involves adjusting the current model to account for the newly calculated tree and its weight. Subsequently, the final model, which is the aggregate of all N trees, is employed to generate predictions for new input data, see Figure 1.4. This

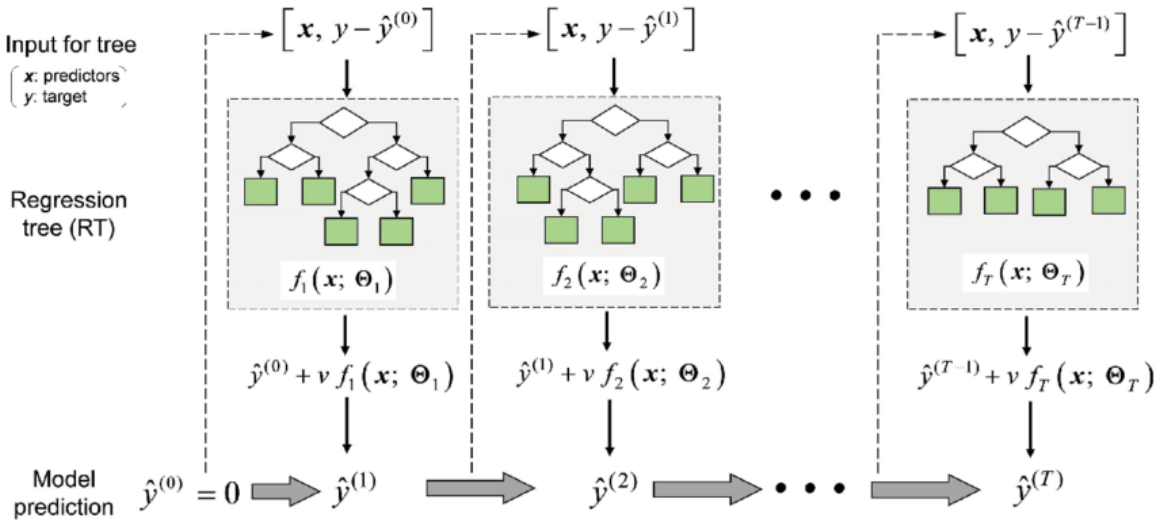

Figure 1.4: Gradient boosting decision tree diagram [Image: Zhang et al., 2023]

approach ensures that each successive tree refines the model by correcting errors from previous iterations, thereby enhancing the overall predictive accuracy of the model.

GBDT is well-known for its simplicity, yet it remains a powerful, widely used, and robust method against overfitting [61]. One notable advantage of GBDT over deep neural networks (DNNs) is its faster training speed, facilitating rapid model development and experimentation, which can be crucial in scenarios requiring swift model iteration and deployment.

XGBoost, or Extreme Gradient Boosting, is an advanced implementation of GBDT that incorporates several enhancements and optimizations. Unlike traditional GBDT, XGBoost includes both L1 (Lasso) and L2 (Ridge) regularization, which helps to prevent overfitting and improve model generalization. Additionally, XGBoost has an inbuilt capability to handle missing data by automatically learning the best imputation strategy. Another key difference is that XGBoost utilizes parallel processing to build trees, making it significantly faster and more efficient than traditional GBDT. XGBoost also employs a more sophisticated tree pruning technique known as "max depth," which improves the accuracy of the model by trimming trees after a specified depth. Furthermore, XGBoost is designed to be highly scalable, capable of handling very large datasets effectively. These enhancements make XGBoost a powerful tool for a wide range of predictive modeling tasks, offering improved performance and efficiency compared to traditional GBDT.

1.3.4 Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNNs) are a class of deep neural networks commonly used for analyzing visual data. They are particularly effective for tasks such as image recognition, object detection, and segmentation due to their ability to capture spatial hierarchies in data. A typical CNN architecture consists of three key layers:

convolutional layers, activation functions, pooling layers, and fully connected layers. Figure 1.5 depicts a typical diagram of CNN architecture.

The convolutional layer is a crucial part of a CNN, where most of the computations occur. In this layer, a filter or kernel—a small matrix of weights—moves across the receptive field of an input image to identify specific features.

The kernel slides over the image's width and height, gradually covering the entire image over several iterations. At each position, it calculates a dot product between the kernel's weights and the pixel values of the image beneath it. This process converts the input image into a series of feature maps or convolved features, each indicating the presence and intensity of particular features at different locations in the image. To maintain the spatial dimensions of the input, padding is often applied to the edges of the image. Padding adds extra pixels around the border of the input image, allowing the kernel to properly cover the edges and preserve the size of the output feature maps.

Mathematically, the convolution operation for a 2D input image I with a kernel K is defined as:

$$(I * K)(i, j) = \sum_{m=-k}^{k} \sum_{n=-k}^{k} I(i + m, j + n) \cdot K(m, n)$$

where k is the kernel size, and (i, j) are the spatial coordinates. This operation involves sliding the kernel over the input image and computing the dot product between the kernel and the overlapping region of the image, producing feature maps that capture local patterns such as edges.

After the convolutional layer, a nonlinear activation function is applied to introduce non-linearity into the model. The most commonly used activation function in CNNs is the Rectified Linear Unit (ReLU), which is mathematically defined as:

$$\text{ReLU}(x) = \max(0, x)$$

The ReLU function sets all negative values in the feature map to zero, allowing the network to learn more complex patterns.

Following the activation layer, a pooling layer is typically applied. Pooling is a downsampling method that reduces the feature dimensions, helping to decrease the computational load and control overfitting. The most common pooling operation is max pooling, which takes the maximum value in each sub-region of the feature map. Mathematically, max pooling with a pool size of $p \times p$ can be defined as:

$$P_{i,j} = \max\{I(m, n) \mid (m, n) \in R_{i,j}\}$$

where $R_{i,j}$ represents the receptive field in the input feature map corresponding to the pooling window centered at (i, j).

Fully connected layers are used at the end of the CNN architecture to make predictions. Each neuron in a fully connected layer is connected to every neuron in the previous layer, allowing the network to combine features learned at different levels of abstraction.

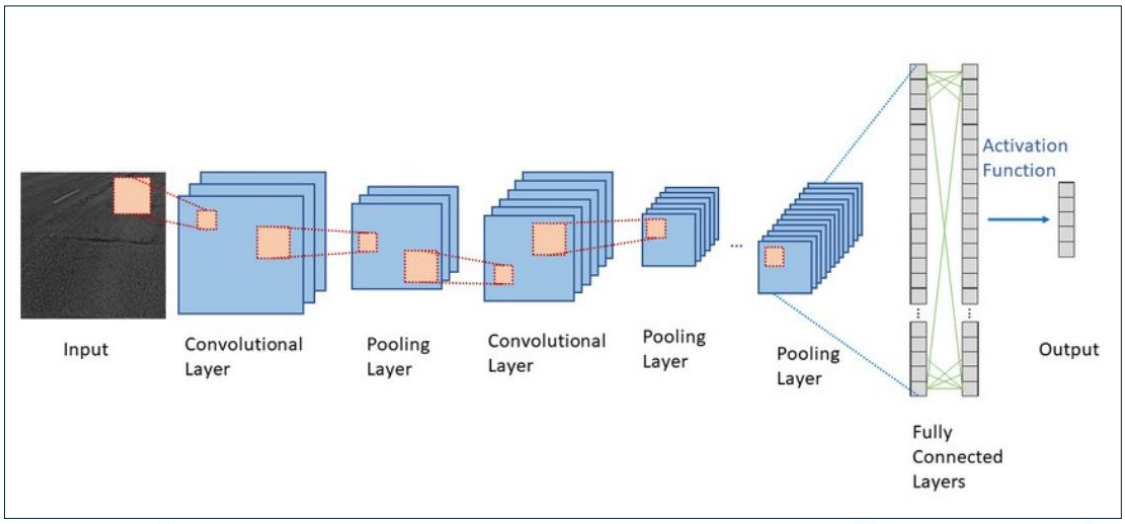

Figure 1.5: A diagram of typical CNN architecture [Image: Moustafa, 2023]

Training a CNN involves minimizing a loss function using an optimization algorithm. The choice of the loss function depends on the type of task. For classification tasks, the *cross-entropy* loss function is commonly used:

$$\text{Cross-Entropy} = -\sum_{i} y_i \log(p_i)$$

where y_i is the true label and p_i is the predicted probability for class i

For regression tasks, the common loss function is *mean squared error (MSE)*, which is the average of the squared differences between the predicted, $\hat{y}_i$, and the actual, y_i as defined by:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Once a loss function is determined, an optimization method is used to minimize this loss function. The optimizer iteratively changes the model parameters (weights and biases) to minimize the specified loss. The most commonly used algorithm for training neural networks is *Stochastic Gradient Descent (SGD)*, and it serves as the foundation for many other optimization methods. SGD works by iteratively updating model parameters based on the gradient of the loss function with respect to those parameters, computed over small, randomly selected batches of data, rather than the entire dataset as defined by

$$\theta_{t+1} = \theta_t - \alpha \nabla_\theta J(\theta_t)$$

where θ_t represents the model parameters at iteration t, α is the learning rate, and $J(\theta_t)$ is the loss function.

Adaptive Moment Estimation (Adam) is another widely used optimizer that utilizes estimates of the first and second moments of the gradients to provide the following parameter update:

$$\theta_{t+1} = \theta_t - \frac{\alpha \hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

where $\hat{m}_t$ is the estimate of the first moment (mean), $\hat{v}_t$ is the estimate of the second moment (uncentered variance) defined by

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

with the estimates of the first and second moments of the gradients,

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \nabla_\theta J(\theta_t)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)(\nabla_\theta J(\theta_t))^2$$

where, β_1 and β_2 are exponential decay rates for these estimates, α is the learning rate, and ϵ is a small constant added for numerical stability.

1.3.5 Graph Convolutional Neural Network (GCN)

The Graph Convolutional Network (GCN) is a type of Graph Neural Network (GNN), presented by Kipf and Welling (2017) [66], specifically designed for semi-supervised classification tasks on graph-structured data. It employs an efficient layer-wise propagation rule based on a first-order approximation of spectral convolutions on graphs. This model has the capability to encode both the graph structure and node features, making it particularly useful for semi-supervised classification tasks.

There are mainly two types of GCNNs: spatial GCNNs and spectral GCNNs. Spatial GCNNs operate directly on the graph structure by defining convolution operations in the spatial domain of the graph. They perform convolutions by aggregating information from a node's neighbors, effectively capturing local graph structures. On the other hand, spectral GCNNs define convolution operations in the spectral domain using the graph Laplacian eigenvectors. This approach transforms the graph signal into the spectral domain, applies filters, and then transforms it back to the spatial domain.

We investigated spatial-based graph convolutional models, similar to the Graph Convolutional Network (GCN) introduced by [66]. The GCN model operates on an undirected graph, $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with N vertices $v_i \in \mathcal{V}$, edges $(v_i, v_j) \in \mathcal{E}$, and $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the weighted adjacency matrix. The graph Laplacian matrix $\mathbf{L}$ is defined as,

$\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D}$ is the degree matrix $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$, along with feature vector matrix $X \in \mathbb{R}^{N \times C}$ assigned to each node with C be the dimension of a feature vector. After receiving the graph input and feature vector matrix A, the GCN proceeds with the task of learning representations of the graph's nodes and their relationships and produces a node feature matrix as follows.

Each neural network layer in a GCN can be expressed as a non-linear function:

$$H^{(l+1)} = f(H^{(l)}, A)$$

where $H^{(0)} = X$, is the initial node feature matrix and $H^{(L)} = Z$ for graph-level outputs), with L being the total number of layers. The difference between specific models lies in how $f(\cdot, \cdot)$ is defined and parameterized. Hence, the layer-wise propagation defined by

$$f(H^{(l)}, A) = \sigma(AH^{(l)}W^{(l)})$$

where $W(l)$ is a weight matrix for the lth neural network layer and $\sigma(\cdot)$ is a non-linear activation function like the ReLU.

There are two limitations of this simple model: multiplying with A results in summing the feature vectors of all neighboring nodes but not the node itself. This limitation can be addressed by adding self-loops, achieved by adding the identity matrix I to A, i.e. $\hat{A} = A + I$.

The second limitation is that A is typically not normalized, which changes the scale of the feature vectors. Normalizing A so that all rows sum to one, i.e., $D^{-1}\hat{A}$, where D is the diagonal node degree matrix, addresses this. Multiplying by $D^{-1}\hat{A}$ corresponds to averaging neighboring node features. In practice, symmetric normalization, i.e., $D^{-1/2}\hat{A}D^{-1/2}$, is used for better results. Combining these adjustments, we arrive at the propagation rule introduced by Kipf and Welling (2017) [66]:

$$f(H^{(l)}, A) = \sigma(D^{-1/2}\hat{A}D^{-1/2}H^{(l)}W^{(l)})$$

where $\hat{A} = A + I$ and $\hat{D}$ is the diagonal node degree matrix of $\hat{A}$.

Following the above, a Graph Convolutional Network (GCN) typically proceeds with post-processing steps that are specific to the task being performed. If the task is classification, the output from the readout phase may be passed through a softmax function to obtain class probabilities. These probabilities can then be used to make predictions for each class. If we consider a two-layer GCN for semi-supervised node classification on a graph with a symmetric adjacency matrix A and if we calculate $\tilde{A} = D^{-1/2}\hat{A}D^{-1/2}$ in the pre-processing step, then the model can be described as:

$$Z = f(X, A) = softmax(\tilde{A}ReLU(\tilde{A}XW^{(0)})W^{(1)})$$

Here, $W^{(0)} \in \mathbb{R}^{C \times H}$ is an input-to-hidden weight matrix for a hidden layer with H feature maps. $W^{(1)} \in \mathbb{R}^{H \times F}$ is a hidden-to-output weight matrix. Both $W^{(0)}$ and $W^{(1)}$ are trained using gradient descent. The softmax activation function, defined as $\text{softmax}(x_i) = \frac{\exp(x_i)}{Z}$ with $Z = \sum_i \exp(x_i)$ is applied row-wise. An illustration of a

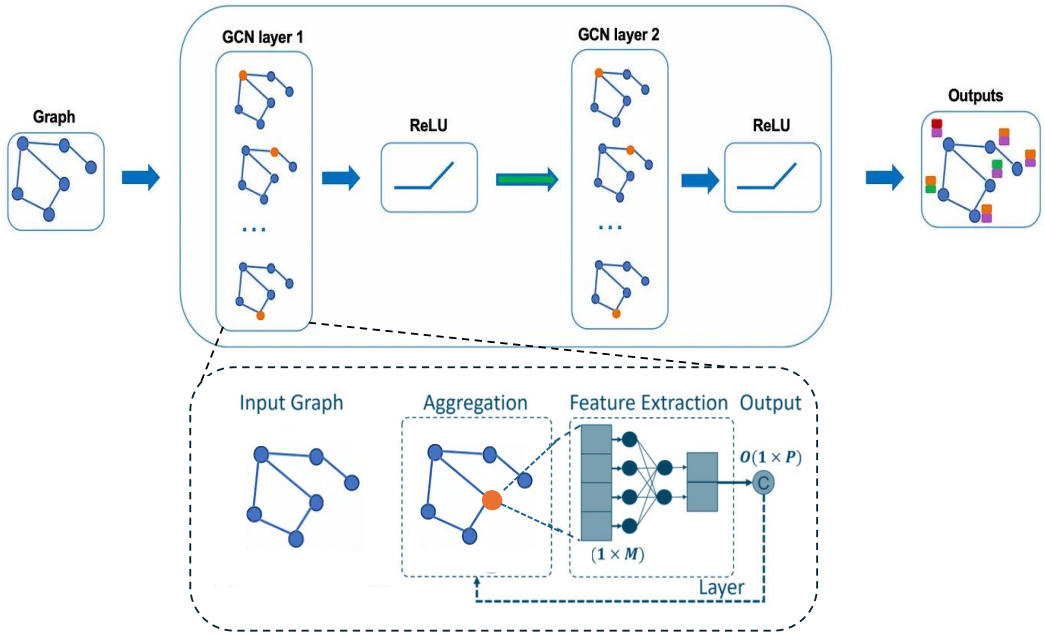

Figure 1.6: A diagram of graph convolutional network with two GCN layers and ReLU activation function [Image: Pham, 2020]

graph convolutional network with two GCN layers and a ReLU activation function has been depicted in Figure 1.6

In computational biology, the structure of GCN lends itself well to analyzing molecular structures represented as graphs [136, 77]. By leveraging the inherent connectivity and spatial relationships between atoms and molecules, GCN can effectively learn representations that capture important molecular features for tasks such as drug discovery, protein-ligand binding prediction, and molecular property prediction.

Chapter 2 Mathematical Graph-Based Machine Learning Models

2.1 Backgrounds

Graph theory, a branch of discrete mathematics established in the 18th century by Leonhard Euler with his solution to the Königsberg bridge problem [39], has evolved significantly over the centuries. It provides a robust framework for modeling and analyzing pairwise relationships through vertices (nodes) and edges (links). The utility of graph theory spans various scientific domains beyond mathematics, including chemistry and biology, where it is particularly valuable for representing and analyzing molecular structures.

In molecular modeling, graph theory offers a natural and intuitive way to depict molecules. Atoms are treated as vertices, while chemical bonds or interactions are considered edges. This representation facilitates the application of graph-theoretic concepts to investigate the structural and functional properties of molecules.

The application of graph theory to chemistry dates back to the early 20th century. Jacobus Henricus van't Hoff and Joseph Achille Le Bel, although not explicitly using graph theory, laid the groundwork for its application in chemistry through their pioneering work in stereochemistry, which involves understanding the spatial arrangement of atoms in molecules [121]. This early work was foundational in showing that the spatial relationships in molecules could be systematically studied and understood, a key concept that graph theory would later formalize.

Subsequently, the mid-20th century saw the development of topological indices, such as the Wiener index, which further demonstrated the potential of graph theory in quantifying molecular properties and predicting chemical behavior. Harold Wiener introduced the Wiener index in 1947 as a way to correlate molecular structure with boiling points of paraffins [134]. Topological indices [49], which are numerical values derived from the graph representation of molecules, have been widely used to describe molecular structure, reactivity, and biological activity. These indices include measures such as the degree distribution, centrality, and connectivity, which provide insights into the stability, reactivity, and overall behavior of molecules [93].

Spectral graph theory, a subfield that studies the properties of graphs through the eigenvalues and eigenvectors of associated matrices (e.g., adjacency and Laplacian matrices), has provided new avenues for molecular analysis. The eigenvalues of these matrices, known as the graph spectrum, encapsulate information about the graph's structure and dynamics, where dynamics refers to the study of how molecular structures, particularly proteins, move and change shape over time.. Spectral graph theory has been particularly influential in understanding vibrational modes, stability, and flexibility of molecular structures, leading to techniques such as normal mode analysis (NMA) and elastic network models (ENM) that leverage spectral properties to study protein dynamics and interactions [8, 137].

By bridging historical developments and modern applications, graph theory has continually enhanced our ability to model and predict molecular behavior, providing

valuable insights that drive advancements in chemistry and biology.

2.1.1 Graph Theory Definitions

In graph theory, a graph is a structure that represents a set of objects where some pairs of objects are connected in a meaningful way. These objects are represented by vertices, and each connection between a pair of vertices is called an edge. In the context of machine learning, each vertex is usually associated with a feature vector that provides descriptive information. This representation allows for effective modeling and analysis of complex networks and relationships across various domains, including social networks, biological systems, and recommendation engines.

Definition 1 (Graph). A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a mathematical structure consisting of a set of vertices $\mathcal{V}$ and a set of edges $\mathcal{E}$, where $\mathcal{E} \subseteq \{(v_i, v_j) \mid v_i, v_j \in \mathcal{V}\}$. Each edge $(v_i, v_j) \in \mathcal{E}$ represents a connection or relationship between the vertices v_i and v_j.

Definition 2 (Complete Graph). A complete graph is an undirected graph in which every pair of distinct vertices is connected by a unique edge. In other words, every vertex in a complete graph is adjacent to all other vertices.

Definition 3 (Bipartite Graph). A bipartite graph is a type of graph where the set of vertices can be divided into two distinct, non-overlapping subsets U and V such that no two vertices within the same subset are adjacent. Formally, a graph $\mathcal{G} = (U, V, \mathcal{E})$ is bipartite if every edge $(u, v) \in \mathcal{E}$ connects a vertex $u \in U$ to a vertex $v \in V$. This means there are no edges connecting vertices within the same subset U or V. Bipartite graphs are often used to model relationships between two different classes of objects, such as jobs and workers, or students and classes.

Definition 4 (Complete Bipartite Graph). A Complete bipartite graph is a special type of bipartite graph where every vertex of one set is connected to every vertex of other sets. That is, a complete bipartite graph $\mathcal{G} = (U, V, \mathcal{E})$ is a graph where the vertex set is partitioned into two subsets U and V such that each vertex in U is connected to every vertex in V, and there are no edges between vertices within the same subset.

Examples of a graph, complete graph, bipartite graph, and complete bipartite graph are illustrated in Figures 2.1a, 2.1b, 2.1c, and 2.1d, respectively.

Definition 5 (Directed and Undirected Graph). A **directed graph** (or digraph) is one where each edge has a specific direction, represented as an ordered pair (u, v), indicating a connection from vertex u to vertex v. In this type of graph, the edge (u, v) is different from (v, u).

In contrast, an **undirected graph** has edges without any direction. Each edge is an unordered pair $\{u, v\}$, indicating a bidirectional connection between vertices u and v. Here, the edge $\{u, v\}$ is considered identical to $\{v, u\}$. Figure 2.2 presents an example of directed and undirected graphs.

Definition 6 (Subgraph). A **subgraph** is a portion of a graph that consists of a subset of the vertices and edges of the original graph. Specifically, if $G = (V, E)$ is a graph, then a subgraph $H = (V_H, E_H)$ is defined such that $V_H \subseteq V$ and $E_H \subseteq E$.

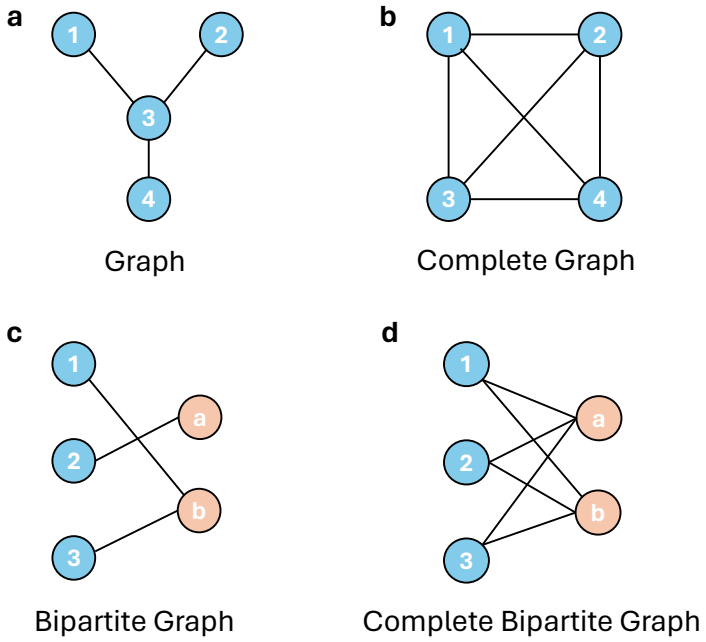

Figure 2.1: A diagram of (a) a simple graph, (b) a complete graph, (c) bipartite graph, and (d) complete bipartite graph.

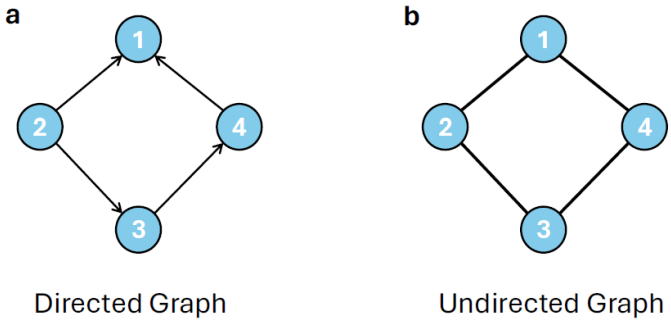

Figure 2.2: A diagram of (a) directed graph, (b) undirected graph.

The subgraph retains the relationships between the vertices as they are in the original graph.

Definition 7 (Colored Graph). A **colored graph** is a graph in which each vertex or edge is assigned a color. Vertex coloring and edge coloring are used to study various graph properties and solve problems such as the graph coloring problem, where the goal is to assign colors to vertices so that no two adjacent vertices share the same color.

Definition 8 (Weighted Graph). A weighted graph is a graph where each edge is

assigned a numerical value known as a weight. These weights can denote different measures depending on the scenario, such as distance, cost, time, or capacity.

Definition 9 (Adjacency Matrix). The **adjacency matrix** of a graph G with n vertices is an $n \times n$ matrix A where $A_{ij} = 1$ if the pairs of vertices i and j are adjacent and $A_{ij} = 0$ otherwise. For undirected graphs, the adjacency matrix is symmetric.

Definition 10 (Degree Matrix). The **degree matrix** D of a graph is a diagonal matrix where each diagonal element D_{ii} represents the degree of vertex i. The degree of a vertex is the number of edges incident to it.

Definition 11 (Laplacian Matrix). The **Laplacian matrix** L of a graph $\mathcal{G}$, also called a graph Laplacian, is defined as $L = D - A$, where D is the degree matrix and A is the adjacency matrix of the graph. The Laplacian matrix is used in various applications, including spectral graph theory and network analysis.

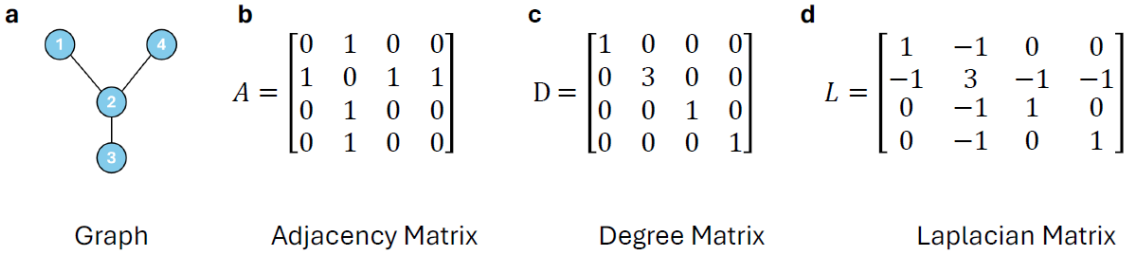

Figure 2.3: A diagram of a) a graph with four vertices, b) the corresponding graph adjacency matrix, c) the degree matrix, and, d) the graph Laplacian matrix.

Mathematically, the Laplacian matrix L of a graph exhibits several key properties that are fundamental to its application in spectral graph theory.

- L is symmetric and positive semi-definite, implying that all of its eigenvalues are non-negative if the graph is undirected.
- The sum of the elements in each row (and column) of L is zero. This property is due to the fact that each row sum in A is equal to the corresponding diagonal entry in D.
- The eigenvalues of L are of particular interest in spectral graph theory. The smallest eigenvalue of L is always 0, corresponding to the eigenvector $\mathbf{1}$ (the all-ones vector). The second smallest eigenvalue, known as the algebraic connectivity or Fiedler value, provides information about the graph's connectivity. This eigenvalue is greater than 0 if and only if $\mathcal{G}$ is a connected graph. This stems from the principle that the number of zero eigenvalues in the Laplacian matrix corresponds to the number of connected components in the graph. The value of this eigenvalue reflects the overall connectivity strength of the graph.

Figure 2.3 presents an example of a simple graph and its corresponding adjacency, degree, and Laplacian matrices.

2.2 Graph Theory-Based Methods for Biomolecules

RI-Score

In this research, Wei et al. [14] introduce the Multiscale Weighted Colored Graph (MWCG) model to enhance the analysis of protein flexibility and rigidity. This novel approach leverages geometric graph theory to improve predictions of protein structural fluctuations, specifically targeting the Debye-Waller factors or B-factors, which correlate with protein flexibility and function.

The Multiscale Weighted Colored Graph (MWCG) model provides an advanced framework for analyzing protein flexibility and rigidity by employing a geometric graph-based approach. In this model, a protein is represented as a graph $G(V, E)$, where V is the set of vertices (atoms) and E is the set of edges (interactions). Each edge $e_{ij} \in E$ is associated with a weight w_{ij} and a color that corresponds to the type of atomic interaction of a protein-graph (e.g., carbon-carbon (CC), carbon-nitrogen (CN), carbon-oxygen (CO)).

A central concept in this model is the use of generalized centrality measures to assess the rigidity of subgraphs. The rigidity at each node i is quantified using correlation functions that consider multiscale interactions. Specifically, the rigidity measure μ_i for node i is computed as:

$$\mu_i = \sum_{j=1}^{N} w_{ij} \Phi_k(\|r_i - r_j\|; \eta_{ij})$$

where w_{ij} is the weight function, Φ_k is a generalized correlation function, $\|r_i - r_j\|$ is the Euclidean distance between atoms i and j, and η_{ij} is a characteristic distance. The correlation functions used are often generalized exponential or Lorentz functions, such as:

$$\Phi_k(\|r_i - r_j\|; \eta_{ij}) = e^{-\left(\frac{\|r_i - r_j\|}{\eta_{ij}}\right)^{\kappa}}$$

$$\Phi_k(\|r_i - r_j\|; \eta_{ij}) = \frac{1}{1 + \left(\frac{\|r_i - r_j\|}{\eta_{ij}}\right)^{\nu}}$$

Where, κ and τ are exponential and Lorentz kernel parameters ($\kappa = 1$, $\tau = 1, 3$). These functions are chosen to ensure they approach 1 as the distance goes to 0 and 0 as the distance goes to infinity, reflecting the strength of interactions based on distance.

The flexibility index f_i is then derived as the reciprocal of the rigidity measure:

$$f_i = \frac{1}{\mu_i}$$

This flexibility index is used to predict the B-factors of protein atoms, which correlate with thermal motion. The theoretical B-factor B_i for atom i is modeled as:

$$B_i^t = \sum_{k=1}^{m} c_k f_i^k + b$$

where c_k are coefficients, f_i^k are flexibility indices at different scales, and b is a constant offset. These coefficients are determined by minimizing the error between the theoretical and experimentally measured B-factors B_i^e:

$$\min_{c_k,b} \sum_{i=1}^{N} \left(B_i^t - B_i^e\right)^2$$

To incorporate multiscale interactions, the model uses multiple characteristic length scales η^n to capture short-range, medium-range, and long-range interactions. The flexibility of atom i at scale n due to interaction type k is given by:

$$f_i^{k,n} = \frac{1}{\sum_{j=1}^{N} w_{ij}^n \Phi_k(\|r_i - r_j\|; \eta_{ij}^n)}$$

where η_{ij}^n is the scale-specific characteristic distance. The final B-factor prediction integrates these multiscale flexibility indices.

The MWCG model has been validated on diverse protein datasets, demonstrating superior accuracy compared to traditional methods like Normal Mode Analysis (NMA) and the Gaussian Network Model (GNM). The Pearson correlation coefficients for MWCG often exceed 0.8, significantly improving the reliability of protein flexibility predictions.

In applications, the MWCG model provides detailed predictions of B-factors and flexibility indices for all atoms in a protein, enhancing structural analysis and insights into protein function and stability. Its ability to account for multiscale interactions makes it a powerful tool for understanding complex molecular dynamics.

AGL-Score

The AGL-Score model, developed by Nguyen and Wei (2019) [93], represents an advanced approach in predicting protein-ligand interactions, focusing on scoring, ranking, docking, and screening. This model leverages algebraic graph theory to encode high-dimensional physical and biological information into low-dimensional representations using multiscale weighted colored subgraphs. A subgraph is a smaller graph extracted from the original graph that focuses on interactions between specific types of atom elements such as C-O, C-N, N-O so on. These subgraphs allow the model to handle the complexity of different interaction types separately and at multiple scales.

The initial step in the AGL-Score model involves constructing a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ for each molecule or a biomolecular complex. In this graph, atoms are represented as vertices, and interactions between atoms are represented as edges. Each vertex $\mathcal{V}$ is

labeled with its corresponding element type α_j and its position r_j.

$$\mathcal{V} = \{(\mathbf{r}_j, \alpha_i) | \mathbf{r}_j \in \mathbb{R}^3; \alpha_j \in \mathcal{C}; j = 1, 2, \cdots, N\} \tag{2.1}$$

The interactions (edges) $\mathcal{E}$ between these atoms are quantified using weight functions, which capture the strength of the interaction based on the distance between atoms and their types.

$$\begin{aligned} \mathcal{E} = \{&\Phi(\|\mathbf{r}_i - \mathbf{r}_j\|; \eta_{kk'}) | \alpha_i = \mathcal{C}_k,\ \alpha_j = \mathcal{C}_{k'}; \\ &i, j = 1, 2, \cdots, N;\ \|\mathbf{r}_i - \mathbf{r}_j\| > r_i + r_j + \sigma\}, \end{aligned} \tag{2.2}$$

The subgraph weight function Φ is designed to decay with increasing distance between atoms, ensuring that closer interactions are weighted more heavily. Mathematically, this function is defined such that it approaches 1 as the distance approaches zero and decays toward zero as the distance increases. This allows the model to effectively capture non-covalent interactions that are crucial for protein-ligand binding.

$$\begin{aligned} &\Phi(\|\mathbf{r}_i - \mathbf{r}_j\|; \eta_{kk'}) = 1, \quad \text{as } \|\mathbf{r}_i - \mathbf{r}_j\| \to 0, \\ &\Phi(\|\mathbf{r}_i - \mathbf{r}_j\|; \eta_{kk'}) = 0, \quad \text{as } \|\mathbf{r}_i - \mathbf{r}_j\| \to \infty, \alpha_i = \mathcal{C}_k,\ \alpha_j = \mathcal{C}_{k'}. \end{aligned} \tag{2.3}$$

Once the subgraphs are constructed, the next step is to derive algebraic representations using matrices such as the Laplacian matrix (L) and the adjacency matrix (A) to encode the connectivity and interaction strengths within the graph.

The eigenvalues λ_j and eigenvectors u_j of these matrices are computed to derive descriptors that represent the molecular properties. The eigenvalues provide a spectrum that captures various interaction strengths and connectivity patterns within the graph. For instance, an atomic descriptor μ_i for the $i-th$ atom can be calculated as:

$$\mu_i^L(\eta_{kk'}) = \sum_l (u_{il}^L)^2 \lambda_l^L \tag{2.4}$$

Statistical measures of these nontrivial eigenvalues, such as their sum, mean, maximum, minimum (the Fiedler value, that is the first non-zero eigenvalue of $L(\eta_{kk'}$ is the algebraic connectivity or Fiedler value, which is included as the smallest eigenvalue), and standard deviation, are used as features in the predictive model. These measures capture different aspects of the molecular structure and interactions, providing a comprehensive representation.

The AGL-Score model has demonstrated superior performance across multiple PDBbind benchmark datasets, outperforming other state-of-the-art scoring functions in scoring, ranking, docking, and screening tasks. This success validates the model's ability to accurately capture and predict complex molecular interactions using algebraic graph theory and machine learning. The combination of detailed graph-based representations and robust machine learning techniques makes the AGL-Score model a powerful tool for protein-ligand binding analysis.

2.3 Algebraic Graph Learning-based Machine Learning Model for Extended Atom Type (AGL-EAT)

2.3.1 Model Development

Extended Atom-Type Multiscale Weighted Colored Subgraphs

In this section, we explore the development of comprehensive graph theory descriptors for a biomolecule or molecular complex. A biomolecular graph, denoted as $\mathcal{G}(\mathcal{V}, \mathcal{E})$, is composed of vertices $\mathcal{V}$ and edges $\mathcal{E}$, providing a powerful tool for representing non-covalent interactions among atoms within the molecule. This graph theory representation is further enriched through the technique of graph coloring, which assigns distinct labels to various types of elements. This coloring process creates a graph that encodes different atomic interactions, enabling the construction of an inclusive and simplified representation of the dataset. Within this framework, atoms in the molecule, identified by these labels, are organized into subgraphs, and the colored edges signify element-specific interactions.

In our previous studies [94, 93], the classification of interactions relied on combinations of element symbols of protein-ligand atoms involved, such as C–O, C–N, etc. Following our most recent work[102], bipartite-colored subgraphs are defined for protein-ligand complexes, where graph coloring is based on extended atom types for proteins and SYBYL atom types for ligands. Protein atom types are identified by their names within the protein data bank (PDB) structure such as carbon alpha (CA), carbon beta (CB), carbon delta-1 (CD1), etc. These atom names serve as identifiers for specific positions within a protein's three-dimensional arrangement. They help define the individual atoms that constitute amino acids, the building blocks of proteins, and provide crucial information about their spatial arrangement and chemical properties. A total of 37 distinct atom names are considered that are frequently found in protein structures within the PDB database. In ligand Tripos Mol2 structure, SYBYL atom types classify atoms based on their chemical attributes and surroundings within molecular structures, aiding in the identification of diverse atom categories, considering factors like hybridization state, bonding characteristics, and potential interactions. The incorporation of SYBYL atom types enables precise classification, including distinct subtypes for Carbon (C) elements, such as C.1, C.2, C.3, C.ar, and C.cat. The set $\mathcal{A}_p$ represents atom names of proteins,

$$\mathcal{A}_p = \{C, CA, \ldots, N, ND1, \ldots, O, OD1, \ldots, SD, SG\} \tag{2.5}$$

And, the set $\mathcal{A}_l$ represents atom types of ligands,

$$\mathcal{A}_l = \{As, B, Be, \ldots, C.1, C.2, \ldots, N.1, N.2, \ldots, V, Zn\} \tag{2.6}$$

For convenience, We define $\mathcal{A}$ as the collection of all atom types within a given molecular dataset as described above, where $\mathcal{A}_k$ denotes the atom type at the ith position within the set. We further symbolize the subgraph vertices as $\mathcal{V}$, which are characterized by the atom coordinates r_i and their corresponding atom types α_i:

$$\mathcal{V} = \{(\mathbf{r}_i, \alpha_i) | \mathbf{r}_i \in \mathbb{R}^3; \alpha_i \in \mathcal{A}; i = 1, 2, \cdots, N\} \tag{2.7}$$

Additionally, we symbolize the subgraph edges as $\mathcal{E}$ and defined as follows:

$$\begin{aligned}\mathcal{E} = \{&\Phi(\|\mathbf{r}_i - \mathbf{r}_j\|; \eta_{kk'})|\alpha_i = \mathcal{A}_k,\ \alpha_j = \mathcal{A}_{k'};\\ &i, j = 1, 2, \cdots, N;\ \|\mathbf{r}_i - \mathbf{r}_j\| \leq c\},\end{aligned} \tag{2.8}$$

Here, we calculate the edge weights based on the characteristics distance $\eta_{kk'}$ between pairs of atom types $\mathcal{A}_k$ and $\mathcal{A}_{k'}$ using the subgraph weight function Φ. And, $\|\mathbf{r}_i - \mathbf{r}_j\|$ denotes the Euclidean distance between the ith and jth atoms and c is a predefined cutoff distance that defines the binding site of the atom type $\mathcal{A}_k$ and $\mathcal{A}_{k'}$. The weight function Φ assesses the interaction strength between atoms, taking into account their Euclidean distances, and it satisfies the following conditions:

$$\begin{aligned}&\Phi(\|\mathbf{r}_i - \mathbf{r}_j\|; \eta_{kk'}) = 1, \quad \text{as } \|\mathbf{r}_i - \mathbf{r}_j\| \to 0,\\ &\Phi(\|\mathbf{r}_i - \mathbf{r}_j\|; \eta_{kk'}) = 0, \quad \text{as } \|\mathbf{r}_i - \mathbf{r}_j\| \to \infty, \alpha_i = \mathcal{A}_k,\ \alpha_j = \mathcal{A}_{k'}.\end{aligned} \tag{2.9}$$

Often, a popular selection for Φ is the generalized exponential function or the generalized Lorentz function denoted as follows:

$$\Phi_E(\|\mathbf{r}_i - \mathbf{r}_j\|; \eta_{kk'}) = e^{-(\|\mathbf{r}_i - \mathbf{r}_j\|/\eta_{kk'})^{\kappa}}, \quad \kappa > 0, \tag{2.10}$$

and

$$\Phi_L(\|\mathbf{r}_i - \mathbf{r}_j\|; \eta_{kk'}) = \frac{1}{1 + (\|\mathbf{r}_i - \mathbf{r}_j\|/\eta_{kk'})^{\nu}}, \quad \nu > 0, \tag{2.11}$$

The generated weighted colored subgraph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ offers a robust depiction of molecular attributes at the atomic scale. Analyzing this subgraph allows us to extract detailed molecular descriptors and explore the system's multiscale behavior. This behavior is a result of considering various characteristic distances $\eta_{kk'}$ for different atom type pairs, enabling the creation of diverse and scalable graph-based descriptors, including the geometric subgraph centrality. This extends the concept of the bipartite subgraph we utilized in our prior research on predicting protein-ligand binding affinities and free energy ranking [94], where every edge connects an atom in the protein with an atom in the ligand. The matrix representations of such subgraphs offer a simple and expressive way to describe the interactions between subgraph elements using matrices. As following our previous study[93], we utilized two highly significant matrices: the Laplacian matrix and the adjacency matrix in our present work.

For each atom-type pair subgraph, we compute the Laplacian matrix $L(\eta_{kk'})$, which is defined as follows:

$$L_{ij}(\eta_{kk'}) = \begin{cases} -\Phi(\|\mathbf{r}_i - \mathbf{r}_j\|; \eta_{kk'}) & \text{if } i \neq j, \alpha_i = \mathcal{A}_k, \alpha_j = \mathcal{A}_{k'} \\ & \text{and } \|\mathbf{r}_i - \mathbf{r}_j\| \leq c; \\ -\sum_j L_{ij} & \text{if } i = j \end{cases} \tag{2.12}$$

The proposed atom-type weighted labeled Laplacian matrix possesses symmetry, diagonal dominance, and positive semidefiniteness, which guarantees that all its eigenvalues are nonnegative. The first non-zero eigenvalue of $L(\eta_{kk'})$ is referred to as the

algebraic connectivity or Fiedler value of the subgraphs. Let us define the eigenvalues and eigenvectors of $L(\eta_{kk'})$ as ${\lambda_j}^L, j = 1, 2, \ldots$. We can directly create a set of weighted labeled Laplacian matrix-based molecular descriptors for extended atom types by utilizing the statistics of nontrivial eigenvalues ${\lambda_j}^L, j = 1, 2, \ldots$. In this context, the Fiedler value is included as the smallest eigenvalue.

For each atom-type pair subgraph, we also compute the adjacency matrix $A(\eta_{kk'})$. We obtain the adjacency matrix by setting the diagonal elements of the Laplacian matrix to zero. The adjacency matrix simplifies the representation while maintaining the fundamental properties of the molecular structure. The adjacency matrix is defined as:

$$A_{ij}(\eta_{kk'}) = \begin{cases} -\Phi(\|\mathbf{r}_i - \mathbf{r}_j\|; \eta_{kk'}) & \text{if } i \neq j, \alpha_i = \mathcal{A}_k, \alpha_j = \mathcal{A}_{k'} \\ & \text{and } \|\mathbf{r}_i - \mathbf{r}_j\| \leq c; \\ 0 & \text{if } i = j \end{cases} \tag{2.13}$$

The adjacency matrix $A(\eta_{kk'})$ is a symmetric, nonnegative matrix that contains the same information as the corresponding Laplacian matrix. However, its eigenvalues λ_j^A $(j = 1, 2, \ldots)$ and eigenvectors u_j^A $(j = 1, 2, \ldots)$ exhibit different behaviors compared to those of the Laplacian matrix.

The Laplacian and adjacency matrices corresponding to the weighted colored subgraph $G_{O.2-N.am}$ of the Xanthin molecule $C_5H_4N_4O_2$ has been illustrated in Figure 2.4.

Indeed, eigenvalue analysis is widely recognized as a computationally expensive task. However, our AGL-EAT-Score approach benefits from two crucial factors that enhance its computational efficiency. Firstly, we restrict matrix constructions to encompass solely those atoms located in the proximity of the protein-ligand binding site. Furthermore, our atom-type-specific criteria further narrow down the atoms involved in each matrix construction. Consequently, we work with numerous small matrices, which enables an efficient spectral approach for analyzing protein-ligand binding affinities by significantly reducing the computational burden and helps maintain the stability of eigenvalue calculations.

2.3.2 Algebraic Graph Learning

We employed Machine learning algorithms to analyze the eigenvalue statistics descriptors generated from the weighted colored subgraph Laplacian matrix or adjacency matrix as discussed above. These extracted features contain valuable information about the complex, allowing the algorithms to identify patterns and predict essential properties of molecular complexes more accurately.

Supervised machine learning algorithms can encompass both classification and regression tasks, the labeled dataset is divided into two subsets: a training set and a test set. Let us denote $\mathcal{G}(\mathcal{X}_i, \lambda)$, a function encoding the geometric information of a molecule into suitable graph representations using $\mathcal{X}_i$, a labeled dataset corresponding to the ith data point in the training set and λ, a set of kernel parameters. The following loss minimization problem further reformulates the optimization process

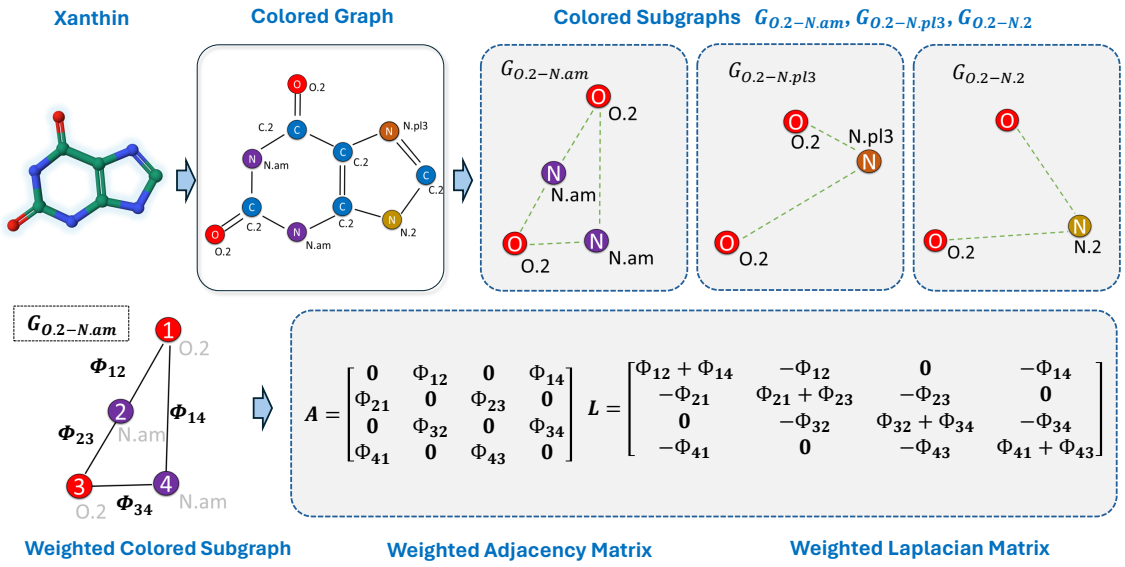

Figure 2.4: An illustration of the construction of weighted adjacency and weighted Laplacian matrices from the weighted colored subgraphs of a molecule. In the top row, an example molecule, Xanthin ($C_5H_4N_4O_2$; CHEBI:17712), the colored graph structure of the xanthin, and three example colored subgraphs of $G_{O.2-N.am}$, $G_{O.2-N.pl3}$, $G_{O.2-N.2}$. In the bottom row, generated weighted adjacency matrix (A) and weighted Laplacian matrix (L) from a example subgraph $G_{O.2-N.am}$.

for training a machine learning model,

$$\min_{\lambda,\theta} \sum_{i \in I} \mathcal{L}(\mathbf{y}_i, \mathcal{G}(\mathcal{X}_i, \lambda); \theta) \tag{2.14}$$

Here, $\mathcal{L}$ denotes a scalar loss function that needs to be minimized, and $\mathbf{y}_i$ refers to the labels assigned to the ith sample in the training set I. The set θ includes hyperparameters that are dependent on the chosen machine learning algorithm and are usually tuned to achieve optimal performance. While various machine learning algorithms, including random forest, gradient boosting trees, graph neural networks, and convolutional neural networks, can be applied alongside the graph descriptors introduced in our study, our primary focus is to assess the effectiveness of the proposed algebraic graph features.

2.4 Geometric Graph Learning-based Ligand-only Machine Learning Model for Extended Atom Type

2.4.1 Model Development

Following the previous section (2.2), our geometric graph learning-based model is developed based on the extended atom type weighted colored subgraphs of the molecules. However, this model was constructed based solely on ligand structures, unlike the geometric graph learning for protein-ligand complexes [102]. We incorporate SYBYL atom types for the ligand structures to integrate the graph coloring based on atom types and their chemical environments, which enhances the representation of molecular structures in our model.

The model begins by constructing a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ representation of the ligand molecule, where $\mathcal{V}$ is the set of vertices and $\mathcal{E}$ is the set of edges. Each atom in the ligand is treated as a node, and the interactions or bonds between these atoms are represented as edges. The graph is then enhanced through the process of graph coloring, wherein different types of atoms are assigned distinct labels. This coloring process helps encode various chemical properties and interaction types into the graph structure. Following our AGL-EAT-Score model, we have employed the SYBYL atom types for the ligands to capture the detailed chemical description. In the Tripos Mol2 structure for ligands, SYBYL atom types categorize atoms based on their chemical properties and the context within molecular structures. This classification aids in identifying various atom types by considering factors such as hybridization state, bonding characteristics, and potential interactions. This comprehensive approach facilitates accurate modeling and analysis of molecular behavior. For instance, there are five subtypes for Carbon (C) elements, five subtypes for Oxygen (O) elements, and seven subtypes for Nitrogen (N) elements.

The notation $\mathcal{T}$ is defined as the collection of all atom types within a given molecular ligand dataset. Here, $\mathcal{T}_k$ represents the atom type at the kth position within the set. The vertices of the subgraph, denoted as $\mathcal{V}$, are characterized by the atom coordinates r_i and their corresponding atom types α_i:

$$\mathcal{V} = \{(\mathbf{r}_i, \alpha_i) | \mathbf{r}_i \in \mathbb{R}^3; \alpha_i \in \mathcal{T}; i = 1, 2, \cdots, N\} \tag{2.15}$$

The edges of the subgraph, symbolized as $\mathcal{E}$, are defined by::

$$\begin{aligned} \mathcal{E} = \{\Phi(\|\mathbf{r}_i - \mathbf{r}_j\|; \eta_{kk'}) | \alpha_i = \mathcal{T}_k,\ \alpha_j = \mathcal{T}_{k'}; \\ i, j = 1, 2, \cdots, N;\ \|\mathbf{r}_i - \mathbf{r}_j\| \leq c\}, \end{aligned} \tag{2.16}$$

In this context, the edge weights are calculated based on the characteristic distance $\eta_{kk'}$ between pairs of atom types $\mathcal{T}_k$ and $\mathcal{T}_{k'}$ using the subgraph weight function Φ. The Euclidean distance between the ith and jth atoms is denoted by $\|\mathbf{r}_i - \mathbf{r}_j\|$. The weight function Φ evaluates the interaction strength between atoms, considering their Euclidean distances, and satisfies the following conditions:

$$\begin{aligned} &\Phi(\|\mathbf{r}_i - \mathbf{r}_j\|; \eta_{kk'}) = 1, \quad \text{as } \|\mathbf{r}_i - \mathbf{r}_j\| \to 0, \\ &\Phi(\|\mathbf{r}_i - \mathbf{r}_j\|; \eta_{kk'}) = 0, \quad \text{as } \|\mathbf{r}_i - \mathbf{r}_j\| \to \infty, \alpha_i = \mathcal{T}_k,\ \alpha_j = \mathcal{T}_{k'}. \end{aligned} \tag{2.17}$$

We utilized a commonly selected form for Φ, specifically the generalized exponential function or the generalized Lorentz function, as defined in Equation (2.9) in this work. These functions provide flexible and robust methods for evaluating the interaction strength between atoms, considering their Euclidean distances and ensuring that the interaction strength appropriately diminishes with increasing distance. This approach allows us to capture the essential characteristics of non-covalent interactions within the molecular structures.

Analyzing this subgraph allows us to extract detailed molecular descriptors and explore the system's multiscale behavior. This behavior is a result of considering various characteristic distances $\eta_{kk'}$ for different atom type pairs, enabling the creation of diverse and scalable graph-based descriptors, including the geometric subgraph centrality, defined as

$$\begin{aligned} \mathrm{GSC}(\eta_{kk'}) = \sum_i \mu_i^G(\eta_{kk'}) = \sum_i \sum_j \Phi(\|\mathbf{r}_i - \mathbf{r}_j\|; \eta_{kk'}), \\ \alpha_i = \mathcal{A}_k,\ \alpha_j = \mathcal{A}_{k'}, \end{aligned} \tag{2.18}$$

This extends the concept of the bipartite subgraph we utilized in our prior research on predicting protein-ligand binding affinities and free energy ranking [94], where every edge connects an atom in the protein with an atom in the ligand. Our objective is to comprehensively characterize the complex by assigning atom-specific descriptions and subgraph weights to the bipartite graph. This approach enables the capture of diverse intermolecular interactions, including hydrogen bonding, polarization, electrostatics, van der Waals interactions, hydrophilicity, and hydrophobicity. Through this method, we aim to develop a comprehensive representation of the complex that considers the unique properties of individual atoms and their interactions. This this work, our objective is to comprehensively characterize the ligand structures by assigning atom-specific descriptions and subgraph weights to the bipartite graph that considers the unique properties of individual ligand atoms and their interactions.

2.4.2 Geometric Graph Learning

The process of combining weighted colored geometric subgraph descriptors with machine learning or deep learning algorithms for predicting molecular properties involves several systematic steps. Weighted colored bipartite geometric subgraphs are constructed for each pair of atom types within the ligand, capturing spatial relationships and interactions while incorporating geometric and structural information.

Once the subgraphs are constructed, machine learning features are generated by aggregating statistical information from each interaction subgraph, including metrics such as the sum, mean, maximum, minimum, and standard deviation. These features collectively provide a comprehensive representation of the complex, independent of molecular specifics. Finally, the generated feature vectors are incorporated into existing machine learning or deep learning algorithms, enabling the utilization of a diverse range of models and techniques to leverage predictive capabilities and strengths inherent in different approaches.

Following this systematic approach, we seamlessly integrate the molecule complex with machine learning or deep learning algorithms to predict molecular properties. The generated features capture relevant information about the complex, enabling the algorithms to learn patterns and make accurate predictions. From the machine learning perspective, we employ a supervised learning algorithm involving a training set and a test set to address either a classification or a regression problem, assuming the dataset is labeled. The training process is formulated as a minimization problem, as shown and discussed in equation (2.10).

Chapter 3 Prediction of Protein-Ligand Binding Affinity

3.1 Backgrounds and Motivations

In the rapidly evolving field of structure-based drug design, the precise prediction of ligand-receptor binding affinity stands as a cornerstone of success. This critical aspect determines the efficacy of a drug (ligand) in interacting with its target, typically a protein, shaping the outcome of drug discovery processes. Central to advancing these predictions is the application of graph theory, a vital branch of discrete mathematics that provides a structured framework for modeling complex relationships in molecular interactions.

Graph theory, with its diverse branches including geometric, algebraic, and topological graph theory, has revolutionized the way scientists approach ligand-receptor binding affinity. Geometric graph theory focuses on spatial connectivity, capturing the essence of geometric relationships within molecular structures [94, 102]. Algebraic graph theory, on the other hand, delves into the realm of algebraic connectivity, utilizing matrix representations like adjacency and Laplacian matrices to elucidate molecular interactions [93, 21]. Topological graph theory bridges graphs and topological spaces, offering insights into the more abstract aspects of molecular configuration [131, 84]. These methodologies have not only enhanced drug discovery but also found widespread applications in biomedical sciences [115, 67, 100], chemical analysis [122, 108, 58, 5], molecular property evaluation [10, 103], and drug repurposing [48, 56].

Scoring functions (SFs) are computational methods used to evaluate protein-ligand interactions and are crucial in structure-based drug design for differentiating between viable and non-viable hypotheses. These scoring functions, based on their theoretical underpinnings, can generally be categorized into the following types: physics-based scoring functions [**?**], empirical scoring functions [145, 127, 36], knowledge-based scoring functions [89, 126], and the increasingly prominent machine learning (ML)-based scoring functions [9, 65]. ML-based SFs, in particular, have garnered attention for their superior performance, driven by extensive datasets, comprehensive molecular descriptors, and advanced machine learning algorithms. However, the efficacy of these functions is often contingent on the size of the training set and the similarity between the training and test sets, a challenge that has been the focus of several recent studies [77, 73, 74, 117, 37, 28].

Significant strides in developing machine learning-based scoring functions have been made by utilizing three distinct types of descriptors. These include physics-based descriptors, which cover aspects like electrostatic binding energies and atomic interactions (Coulombic and van der Waals) [129]; descriptors based on geometric graph theory [94]; and those derived from algebraic topology [17]. The core idea behind these methodologies is the assumption that the essential physical phenomena are typically found within low-dimensional spaces or manifolds, even though they exist in a broader, high-dimensional data space. This concept, while recognized in

the field of manifold learning, presents a major challenge: effectively translating critical physical information from a high-dimensional context into a practical, low-dimensional format for molecules and their complexes. A notable approach to tackle this challenge is the application of multiscale weighted colored subgraphs (MWCS) [14]. In this approach, a protein's structural graph is colored based on the types of interactions between its nodes, leading to the formation of distinct subgraphs. This method stands out for its simplicity, low-dimensional nature, and robustness. A key advantage is its minimal data input requirements for binding affinity predictions, which only need atomic names and coordinates. This simplicity allows the method to circumvent complex data processing and parameterization steps, eliminating the need for molecular mechanical force fields, like charges, bond measurements, van der Waals parameters, and others. This not only streamlines the process but also reduces errors often associated with parameterization.

In our prior research [102], we developed the ^{sybyl}GGL-Score, a sophisticated geometric graph-based method using extended multiscale weighted colored subgraphs for protein-ligand complexes. This approach leveraged graph coloring techniques based on protein atom names and ligand SYBYL atom types. While ^{sybyl}GGL-Score demonstrated exceptional efficacy in predicting protein-ligand binding affinity, surpassing other advanced methods, it did not fully explore the potential of algebraic graph theory within the extended MWCS framework.

In our current study, we take this concept further by developing an algebraic graph-based MWCS with extended atom-type graph coloring, known as the AGL-EAT-Score. This model employs both the Laplacian and adjacency matrices to represent subgraphs, characterizing molecules and their interactions through eigenvalues and eigenvectors. The effectiveness of AGL-EAT-Score has been rigorously evaluated using benchmark datasets like CASF-2016, CASF-2013, and the Cathepsin S dataset. To enhance our model's robustness, we conducted a similarity search to eliminate redundant complexes from our training sets, ensuring a more reliable analysis.

3.1.1 Protein-Ligand Complex

Proteins are molecular devices on the nanometer scale that perform essential biological functions [71]. They are the building blocks of all cells in our bodies and in all living organisms across the world [15]. They are made up of hundreds or thousands of smaller units called amino acids, which are attached to one another in long chains. There are 20 different types of amino acids that can be combined to make a protein. The sequence of amino acids determines each protein's unique 3-dimensional structure and its specific function.

A ligand is a substance that interacts with a biomolecule to perform a specific biological function. Ligands can be ions, small organic molecules, or even macromolecules. In drug discovery, ligands are typically defined as small molecules or atoms that reversibly bind to a receptor, which is a receiving protein molecule. This binding can activate or inhibit the protein's function, thereby influencing biological pathways and cellular responses. The ability of a ligand to bind to a receptor with high specificity and affinity is crucial for its effectiveness as a drug.

The representation of ligands in different dimensions (1D, 2D, and 3D) provides various levels of detail about their structure and properties, which are crucial for understanding their interactions with target biomolecules in drug discovery. The 1D representation refers to the linear depiction of a ligand's chemical structure, typically using its SMILES (Simplified Molecular Input Line Entry System) strings, which represent the structure of the molecule, specifying the types and connectivity of atoms. 2D representation provides a flat depiction of the ligand's structure, showing the arrangement of atoms and the bonds between them. 3D representation provides a three-dimensional view of the ligand, showing the spatial arrangement of atoms and the geometry of the molecule.

A protein-ligand complex is a fundamental component in the field of drug design and molecular biology. It refers to the interaction between a protein molecule, typically an enzyme or receptor, and a ligand, which can be any small molecule, such as a drug candidate or a substrate [6]. This complex is formed when proteins (P) interact with ligands (L), which are molecules that exhibit a high degree of specificity and affinity for binding to the protein. The formation of a protein-ligand complex LP can be described by

$$L + P \rightleftharpoons LP \tag{3.1}$$

Understanding protein-ligand complexes holds significant importance due to their involvement in diverse biological processes and their potential as targets for drug discovery.

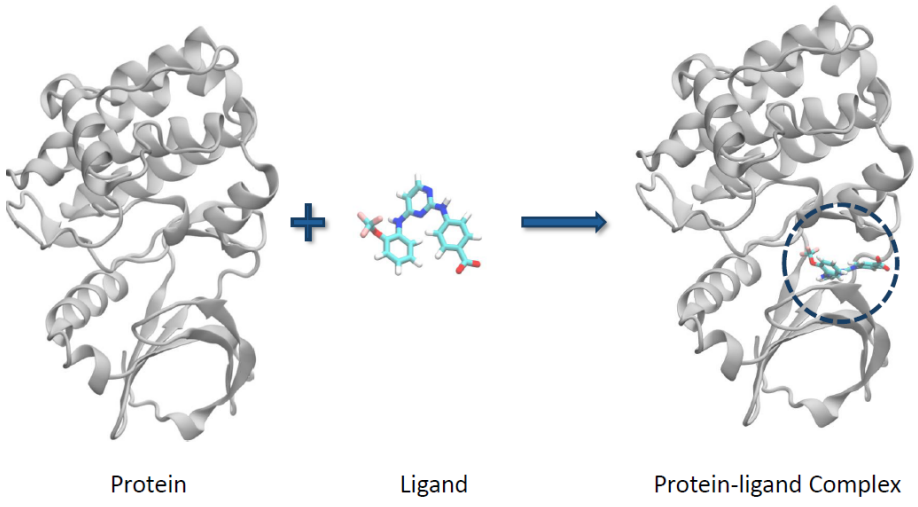

Figure 3.1: A protein-ligand complex with PDBID: 3upp2.

3.1.2 Binding Affinity

The binding affinity between a protein and a ligand is a crucial determinant of the strength and stability of their interaction. It quantifies the strength of the association and reflects how strongly the drug is attracted to the binding site on the target protein. High binding affinity indicates a strong interaction, with the protein and ligand tightly bound together, while low binding affinity suggests a weaker interaction with more transient binding.

Binding affinity, often quantified using various experimental techniques and computational methods. Common parameters for quantifying binding affinity include the equilibrium dissociation constant (Kd), Inhibition Constant (Ki), Association Constant (K_a), half-maximal inhibitory concentration IC_{50}, Gibbs Free Energy (ΔG) so on [59, 11].

Dissociation Constant (K_D): The equilibrium dissociation constant (K_D) is the fundamental parameter used to evaluate the binding properties of a drug to its receptor [4, 120, 109]. Determining K_D values is crucial because it provides essential information about the strength and specificity of the interaction between the drug and its target.

The dissociation constant, k_D, quantifies the equilibrium affinity between a ligand (L) and a protein (P) to form the protein-ligand complex (LP). Mathematically, K_D defined as the ratio of the molar concentrations of the free ligand $[L]$ and the free protein $[P]$ to the concentration of the protein-ligand complex $[LP]$:

$$K_D = \frac{[L][P]}{[LP]} \tag{3.2}$$

It indicates the ligand concentration $[L]$ at which half of the proteins are occupied at equilibrium. In other words, it signifies the ligand concentration at which the number of proteins bound to the ligand $[LP]$ equals the number of proteins with no ligand bound $[P]$. A smaller dissociation constant indicates a tighter binding of the ligand to the protein, implying a higher affinity between them. For instance, a ligand with a nanomolar (nM) dissociation constant exhibits stronger binding to a specific protein compared to a ligand with a micromolar (μM) dissociation constant.

However, researchers often use pK_d, negative logarithm of dissociation constant K_D, defined as

$$pK_D = -\log_{10}(K_D) \tag{3.3}$$

for binding affinity quantification. This is crucial for comparing the affinities and selectivities of ligands. It is also used to predict drug concentrations that are effective in vivo.

Association Constant (K_a): The association constant K_a is the reciprocal of the dissociation constant K_D and is given by:

$$K_a = \frac{1}{K_D} = \frac{[LP]}{[L][P]} \tag{3.4}$$

A higher association constant indicates a stronger affinity between the ligand and the protein.

Inhibition Constant (K_i): The inhibition constant (K_i) is a measure of the potency of an inhibitor in binding to a specific enzyme or receptor. It is defined as the concentration of an inhibitor where the binding to the enzyme is half-maximal. This constant is a crucial parameter in enzyme kinetics and helps in comparing the effectiveness of different inhibitors.

In the context of competitive inhibition, K_i can be determined using the Michaelis-Menten equation modified to include the inhibitor. The general form of the equation for an enzyme-catalyzed reaction with a competitive inhibitor is:

$$v = \frac{V_{\max}[S]}{K_m \left(1 + \frac{[I]}{K_i}\right) + [S]}$$

where, v is the reaction velocity, $V_{\max}$ is the maximum reaction velocity, $[S]$ is the substrate concentration, K_m is the Michaelis constant (the substrate concentration at which the reaction velocity is half of $V_{\max}$), $[I]$ is the inhibitor concentration, K_i is the inhibition constant.

For competitive inhibitors, the relationship between IC_{50} (the concentration of inhibitor where the reaction is half-maximal) and K_i can be described by the Cheng-Prusoff equation:

$$K_i = \frac{IC_{50}}{1 + \frac{[S]}{K_m}}$$

where, IC_{50} is the concentration of the inhibitor that reduces the enzyme activity by 50%, $[S]$ is the fixed substrate concentration, K_m is the Michaelis constant.

For noncompetitive inhibition, IC_{50} is equal to K_i, as the inhibitor binds equally well to both the enzyme and the enzyme-substrate complex, making it independent of the substrate concentration:

$$K_i = IC_{50}$$

In summary, K_i is a critical parameter that quantifies the effectiveness of an inhibitor in enzyme kinetics, with different equations applicable depending on the type of inhibition (competitive, noncompetitive, etc.).

Half-Maximal Inhibitory Concentration (IC_{50}) The half-maximal inhibitory concentration IC_{50} is a measure of the effectiveness of a substance in inhibiting a specific biological or biochemical function. It represents the concentration of an inhibitor where the response (or binding) is reduced by half. IC_{50} is used to compare the potency of various inhibitors. The lower the IC_{50} value, the more potent the inhibitor.

However, IC_{50} is not a direct measure of affinity. Nevertheless, for competitive agonists and antagonists, the relationship between IC_{50} and affinity can be described by the Cheng-Prusoff equation [141]. For enzymatic reactions, this equation is:

$$K_i = \frac{IC_{50}}{1 + \frac{S}{K_m}} \tag{3.5}$$

where K_i is the binding affinity of the inhibitor, IC_{50} is the inhibitory concentration, $[S]$ is the fixed substrate concentration, and K_m is the Michaelis constant, which represents the substrate concentration at which the enzyme activity is half-maximal. For noncompetitive inhibition, IC_{50} is equal to K_i.

Gibbs Free Energy (ΔG) Gibbs free energy is a thermodynamic concept that quantifies the energy available to perform useful work in a system. In the context of protein-ligand interactions, it represents the energy that the drug ligand requires to bind to the protein complex. The Gibbs free energy change ΔG associated with the binding process is related to the dissociation constant by the equation:

$$\Delta G = \Delta G^0 + RT \ln K_D \tag{3.6}$$

where R is the universal gas constant ($8.314\ J/mol \cdot K$), T is the temperature in Kelvin, and ΔG^0 is the standard Gibbs free energy change.

A negative ΔG indicates a spontaneous binding process, meaning the binding of the ligand to the protein occurs favorably. The more negative the ΔG, the stronger the binding affinity, which corresponds to a smaller K_D.

3.2 Methods and Materials

3.2.1 Algebraich Graph Learning for extended Atom Types Methodology

The detailed methodology for our algebraic graph learning (AGL) techniques for molecular complexes is described in Section (2.3). Our study focused on creating comprehensive graph theory descriptors using eigenvalue statistics from weighted colored subgraphs to predict essential molecular properties.

We began by constructing a biomolecular bipartite graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where vertices $\mathcal{V}$ represent atoms and edges $\mathcal{E}$ represent non-covalent interactions among atoms. Each of its edges connects one atom in the protein and another atom in the ligand.. Through graph coloring techniques, distinct labels were assigned to different element types, capturing the intricate interactions within the molecule. This approach resulted in colored subgraphs, each representing specific atomic interactions.

In previous research [94, 93], interaction classifications were based on combinations of element symbols of protein-ligand atoms, such as C–O, C–N, etc. Our recent work [102] expands this by defining bipartite-colored subgraphs for protein-ligand complexes using extended atom types for proteins and SYBYL atom types for ligands.

Protein atom types were identified by names within the protein data bank (PDB) structure, such as carbon alpha (CA), carbon beta (CB), carbon delta-1 (CD1), etc. These atom names served as identifiers for specific positions within a protein's three-dimensional arrangement. Ligand atom types were categorized based on SYBYL atom types, which capture diverse chemical attributes and potential interactions.

For each atom-type pair subgraph, we calculated the Laplacian matrix $L(\eta_{kk'})$ and the adjacency matrix $A(\eta_{kk'})$, using the generalized exponential and Lorentz functions to define the weight function Φ. These matrices provided a comprehensive representation of molecular interactions at the atomic scale. The Laplacian matrix, being symmetric and positive semidefinite, ensured nonnegative eigenvalues, with the smallest nonzero eigenvalue, the Fiedler value, indicating algebraic connectivity. The adjacency matrix, derived by setting diagonal elements of the Laplacian to zero, maintained the fundamental properties of the molecular structure.

Integration with Machine Learning or Deep Learning Algorithms:

Machine learning algorithms were then employed to analyze these eigenvalue statistics descriptors, effectively mapping high-dimensional molecular structures into low-dimensional representations while preserving vital physical and chemical properties. Supervised learning algorithms, including gradient boosting decision trees (GBDTs), were applied for both classification and regression tasks to predict molecular properties. These algorithms were trained and tested on labeled datasets, with the optimization process involving minimizing a loss function $\mathcal{L}$ over the training set, ensuring robust and accurate models.

The labeled dataset was divided into training and test sets, denoted by $\mathcal{G}(\mathcal{X}_i, \lambda)$, where $\mathcal{X}_i$ is the labeled dataset corresponding to the ith data point in the training set and λ is a set of kernel parameters. The following loss minimization problem further reformulates the optimization process for training a machine learning model. The visual depiction of our algebraic graph-based learning approach is presented in Figure3.2.

In our study, the implemented GBDT module in scikit-learn version 0.24.1 with the parameters, `n_estimators` = 20000, `max_depth` = 8, `min_samples_split` = 2, `learning_rate` = 0.005, `loss` = ls, `subsample` = 0.7, and `max_features` = sqrt.

3.2.2 Datasets

To assess the validity and robustness of our proposed model, we employed two widely acknowledged PDBbind benchmark datasets, CASF-2013 and CASF-2016, as well as the CatS dataset, which is part of the D3R Grand Challenge datasets, a global initiative in drug design.

The PDBbind database offers an extensive repository of experimentally determined binding affinity data for biomolecular complexes found within the Protein Data Bank (PDB) [79]. Each PDBbind benchmark dataset is composed of three intersecting subsets: the general set, the refined set, and the core set. The core set is

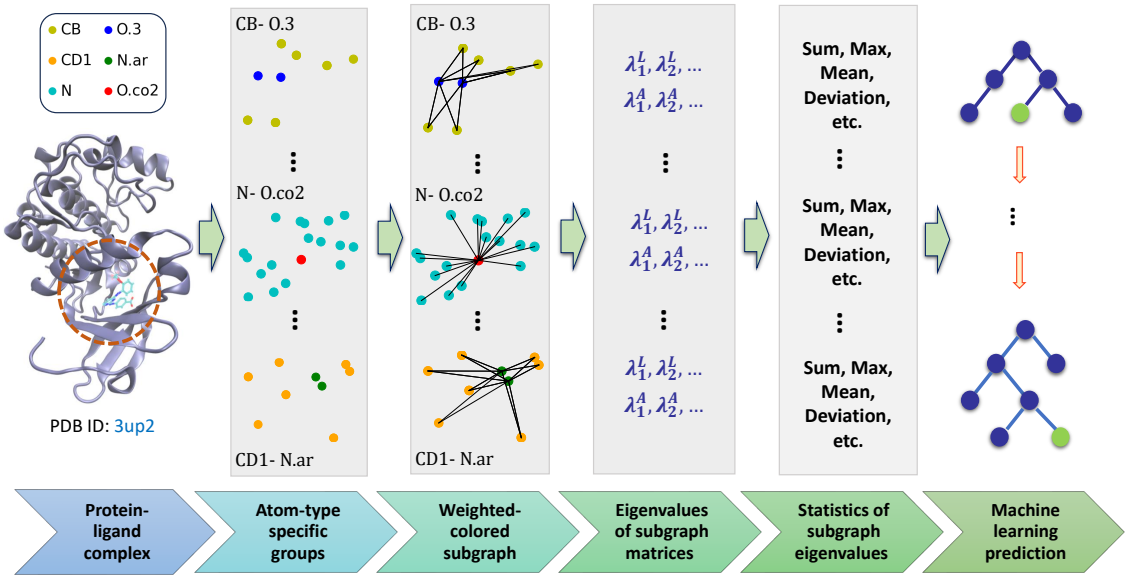

Figure 3.2: Visualization of the AGL-EAT graph learning approach. First column: binding site of the molecular complex identified by its PDBID: 3up2. Second column: three different kinds of protein-ligand atom-type pairs. Third column: weighted colored subgraph representations of the corresponding atom-pairs. Fourth column: eigenvalues of the subgraph Laplacian and adjacency matrices. Fifth column: various statistics of these eigenvalues. Fifth column: advanced machine learning models like gradient boosting trees to combine and process these statistics for training and making predictions.

a subset of both the general set and the refined set and serves as the testing dataset for the respective benchmark.

The general set includes all available data on protein-ligand complexes with experimentally determined binding affinities. It serves as the foundational dataset, providing a broad spectrum of data that captures a wide variety of binding interactions and affinities. This large dataset is primarily used for exploratory data analysis and initial model training, offering a comprehensive view of the binding landscape.

The refined set is derived from the general set and includes complexes that meet stricter quality criteria. These criteria ensure higher resolution crystal structures (better than or equal to 2.5 Å), reliable binding affinity measurements (K_d, K_i), and completeness of both protein and ligand in the structure. The refined set excludes complexes with structural uncertainties or inconsistencies. This subset is crucial for more precise model training and validation, as the higher quality data reduces noise and potential biases, leading to more accurate and reliable models.[22].

The core set is a highly curated subset of both the general and refined sets. It is specifically designed to serve as a benchmark testing dataset. The selection of the

core set involves ensuring diversity in protein and ligand types to prevent overfitting to specific families. It includes only those complexes with the most accurate and reproducible binding affinity measurements, verified to have minimal experimental artifacts. The core set is used for rigorous model evaluation, providing a challenging and unbiased test for predictive models [80]. For more detailed information about the PDBbind datasets, please visit the PDBbind website at http://www.pdbbind.org.cn/.

On the other hand, the Drug Design Data Resource (D3R) Grand Challenge [45, 98] focuses on specific datasets, each dedicated to a single protein and its multiple ligands, all accompanied by experimentally measured affinity data. This setup allows for targeted studies on the binding interactions between a specific protein and various inhibitors or ligands. The Cathepsin S (CatS) dataset, a component of the D3R Grand Challenge, consists of 459 CatS inhibitors. This dataset is specifically designed for binding affinity prediction, providing a focused and detailed set of interactions that are essential for developing and validating predictive models. Detailed information about the CatS dataset can be found on the official D3R website at https://drugdesigndata.org/about/grand-challenge-4/cathepsin_s. A summary of all the datasets used in this study has been listed in Table 3.1 and Table 3.2.

Table 3.1: Summary of PDBbind datasets used to validate our model

Datasets	Training Sets		Test Set
	Refined Set	General Set	CoreSet
CASF-2013 benchmark	3516	11713	195
CASF-2016 Benchmark	3772	12998	285

Table 3.2: Summary of CatS dataset used to validate our model

Dataset	Size of Training Set	Size of Test Set
CatS	431	459

3.2.3 Evaluation Metrics

In this research, we focused on evaluating the performance of our model through two principal metrics: scoring power and ranking power. To determine the scoring power, which is the model's ability to predict binding affinities accurately, we employed Pearson's Correlation Coefficient. This evaluation was conducted using two specific datasets: CASF-2013 [76] and CASF-2016 [118]. Pearson's Correlation Coefficient is a standard method for assessing linear relationships and is ideal for verifying the

precision of the model's predictions in correlation with actual experimental data, as defined by

$$R_p = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}} \tag{3.7}$$

where x_i and y_i are the predicted binding affinity and experimental data for the ith complex, respectively. And, $\bar{x}$ and $\bar{y}$ represent the average of all predicted values and the average of all experimental values in the dataset.

For the assessment of ranking power, which involves the model's capacity to correctly rank interactions based on binding affinities, we utilized both Kendall's Tau and Spearman's Rho.

By definition, Kendall's Tau is a statistical measure used to assess the ordinal association between two variables by evaluating the concordance and discordance of paired data. In the context of ranking, it compares the relative ordering of pairs in the predicted rankings to their true rankings. Ranging from -1 to 1, a higher Kendall's Tau value indicates stronger agreement between predicted and actual rankings. The Kendall's Tau is defined as,

$$\tau = \frac{(n_c - n_d)}{\frac{1}{2}n(n-1)} \tag{3.8}$$

where n_c is the number of concordant pairs (pairs where the order is the same in both predicted and actual rankings), n_d is the number of discordant pairs (pairs where the order differs between predicted and actual rankings), and n is the total number of pairs.

On the other hand, Spearman's Rho, or Spearman's Rank Correlation Coefficient, is a non-parametric measure of the strength and direction of association between two ranked variables. It assesses how well the relationship between two variables can be described using a monotonic function, comparing the ranks of predicted values to actual values. A higher Spearman's Rho value, which ranges from -1 to 1, indicates a stronger correlation and more accurate ranking of predicted outcomes compared to actual results. The Spearman's Rho is defined by,

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)} \tag{3.9}$$

where d_i is the difference between the ranks of each pair and n is the number of pairs.

These evaluations were carried out using the CatS dataset. Kendall's Tau and Spearman's Rho are effective in measuring the ordinal association between predicted outcomes and actual experimental results, thus providing a comprehensive understanding of the model's accuracy in ranking potential interactions.

3.3 Results and Discussion

In this section, we present the results of hyperparameter optimization and the performance of our proposed AGL-EAT-Score on various benchmark datasets. Furthermore, we conduct a rigorous similarity test analysis to validate the robustness of

AGL-EAT-Score. In our study, we adopt the notation AGL-EAT$^{\mathcal{B}}_{\beta,\kappa,\tau}$ to characterize algebraic graph learning for extended atom types features. Here, $\mathcal{B}$ represents the type of matrix used, specifically, $\mathcal{B} = \text{Adj}$ denotes the adjacency matrix, while $\mathcal{B} = \text{Lap}$ refers to the Laplacian matrix. The parameter β indicates the specific kernel types, while κ and τ correspond to the respective kernel parameters. Specifically, we employ the generalized exponential kernel denoted by $\beta = \Phi_E$, and the generalized Lorentz kernel represented by $\beta = \Phi_L$, for generating AGL-EAT features. The parameter τ is a scaling factor determining the characteristic distance $\eta_{kk'}$ between atom type k and atom type k'. This distance is calculated as $\eta_{kk'} = \tau(r_k + r_{k'})$. Here, r_k and $r_{k'}$ are the van der Waals radii of the atoms of type k and type k', respectively.

3.3.1 Hyperparameter Optimization

Hyperparameter optimization of a machine learning model involves the search for the most suitable combination of hyperparameter values that yield the best performance on a specific dataset, all within a reasonable time frame. To make our AGL-EAT-Score model work best for each benchmark, we optimize two key parameters κ and τ for a given kernel type β and matrix type $\mathcal{B}$ for the cutoff distance $c = 12$Å. We use a five-fold cross-validation (CV) along with a grid search method to find the best values for τ, which we search within the range of 0.5 to 10, and for κ, within the range of 0.5 to 20. We increment both parameters by 0.5 in the search, which makes 1600 parameter choices in total. Higher values for the power parameter κ are chosen to approximate the ideal low-pass filter (ILF) [94]. Moreover, there are more than 1000 protein-ligand atomic interactions considering the atom-names of protein and SYBYL atom-type for ligand. We focus on the positive eigenvalues of adjacency matrices while noting that Laplacian matrices are inherently positive semidefinite. From the resulting set of eigenvalues or their corresponding atomic descriptors, we derive nine descriptive statistical values: the sum, minimum, maximum, mean, median, standard deviation, and variance. Additionally, we also include the number of eigenvalues and the sum of the squares of the eigenvalues in our analysis.

Cross-validation is a robust statistical method used to estimate the skill of machine learning models. In five-fold cross-validation, the dataset is randomly partitioned into five equal-sized subsets. The model is trained on four subsets and tested on the remaining subset. This process is repeated five times, each time with a different subset as the test set, and the average performance is taken as the final model evaluation metric. This approach helps ensure that the model's performance is not dependent on the particular division of data and provides a more generalized evaluation.

We applied a five-fold cross-validation on the refined set excluding the core set to find the optimized kernel parameters for each of the PDBbind benchmark datasets. We train our model on both the PDBbind refined set and general set using the derived optimized hyperparameters and then evaluate the model's performance on the corresponding test set. However, for the CatS dataset, we perform a five-fold cross-validation on the training set and evaluate the model's performance on the CatS test set provided by the D3R database.

A detailed discussion of optimized hyperparameters and the model's performances on each of the datasets used in this study has been documented in Figures 3.3,3.5, and 3.6. For the CASF-2016 benchmark dataset, the best models are obtained to be AGL-EAT$^{\text{Adj}}_{\Phi_E,16.5,3.0}$ and AGL-EAT$^{\text{Lap}}_{\Phi_E,19.5,2.5}$ as presented in Figure 3.3 a and b. The median Pearson's correlation coefficient (out of 50 runs) is $R_p = 0.796$ and R_p of 0.795 for the best models reported. According to the five-fold CV performances presented in Figure 3.5, the best models of the CASF-2013 benchmark dataset are AGL-EAT$^{\text{Adj}}_{\Phi_E,5.5,2.0}$ and AGL-EAT$^{\text{Lap}}_{\Phi_E,4.5,2.0}$ with optimized exponential kernel parameters $\kappa = 5.5$, $\tau = 2.0$ with the Adjacency matrix and $\kappa = 4.5$, $\tau = 2.0$ with the Laplacian matrix. The median $R_p = 0.795$ and $R_p = 0.796$ for the corresponding best models with the optimal kernel parameters. Finally, Figure 3.6 illustrates the five-fold CV performances for CatS dataset. The best models for this dataset are AGL-EAT$^{\text{Adj}}_{\Phi_E,5.5,2.0}$ and AGL-EAT$^{\text{Lap}}_{\Phi_E,4.5,2.0}$ with median Kendall's $\tau = 0.57837$ and 0.57305 respectively.

3.3.2 CASF-2016 benchmark

We utilize the CASF-2016 benchmark dataset from the PDBbind database as our initial dataset. Figure 3.3 captures CV results for the CASF-2016 benchmark for the SYBYL atom-type model. For this benchmark, the optimal kernel parameters with the Adjacency matrix are $\beta = \Phi_E$, $\kappa = 16.5$, and $\tau = 3.0$ with a median Pearson's correlation coefficient $R_p = 0.796$ and the optimal kernel parameters with the Laplacian matrix are $\beta = \Phi_E$, $\kappa = 19.5$, and $\tau = 2.5$ with a median R_p of 0.795.

After the best models have been identified for each benchmark, our goal is to assess their performance on the test set by calculating Pearson's correlation coefficient between the predicted and experimental binding affinities. We first train each model using the refined set, and then make predictions on the test set. To ensure reliable predictions, we repeat this process up to 50 times and compute the average of all predicted values to obtain the final predicted set. Next, we train the model using the general set, excluding the CASF-2016 core set. Training on this larger dataset, despite the lower quality 3D structures, will validate the robustness of the proposed models against more diverse and potentially irrelevant data. Similarly, we repeat the model training 50 times to generate predicted values, and these values are then averaged to produce the final prediction.

A summary of the performances of the best AGL-EAT models on the CASF-2016 is presented in Table 3.3. The best model reported is AGL-EAT$^{\text{Adj}}_{\Phi_E,16.5,3.0}$ with $R_p = 0.873$. A comparison within the CASF-2016 benchmark is presented in Figure 3.4b, showcasing our model's superior performance as it ranks at the top among other models [118, 116, 130]. It is important to emphasize that the base geometric and algebraic graph learning models, which account for element-specific interactions rather than atom-type interactions, demonstrate relatively lower performance, with Pearson's correlation coefficients of 0.815 [94] and 0.835 [93], respectively. The comparison presented and illustrated in Figure 3.4b underscores the significant improvement in scoring power and effectiveness when atom-type pair interactions are incorporated into the current model.

Table 3.3: Performance of various AGL-EAT-Score models on CASF–2016 test set.

Model	Trained with Refined Set	Trained with General Set
AGL-EAT$^{\text{Adj}}_{\Phi_E,16.5,3.0}$	0.835	0.873
AGL-EAT$^{\text{Lap}}_{\Phi_E,19.5,2.5}$	0.837	0.871

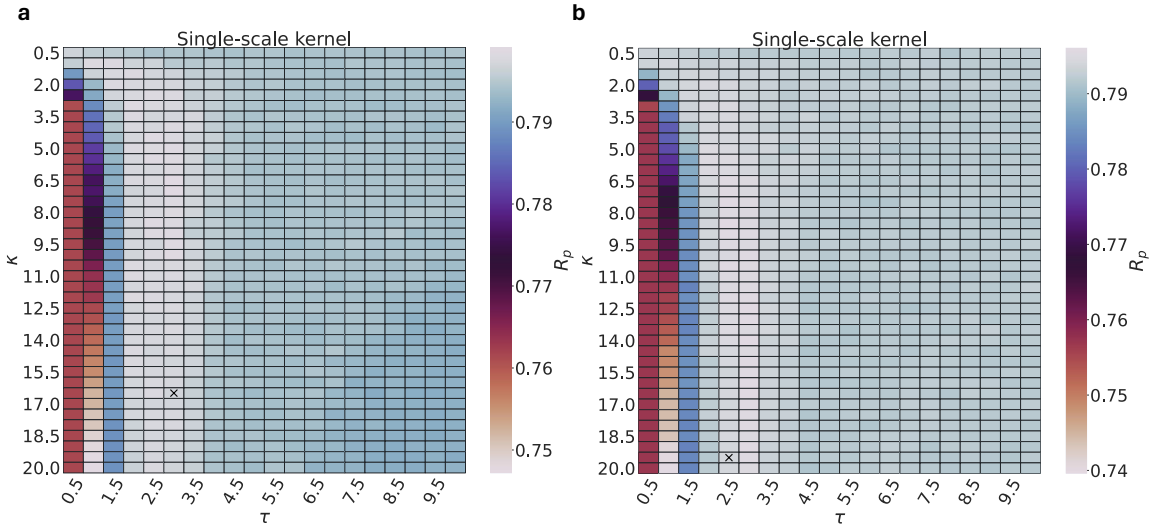

Figure 3.3: The optimized kernel parameters for the algebraic graph learning model with multiple atom types on the CASF-2016 dataset are indicated by 'x' marks, representing the best parameter. The optimal parameters for (a) single-scale model with the Adjacency matrix are $\kappa = 16.5$ and $\tau = 3.0$ with a median Pearson's correlation coefficient $R_p = 0.796$ and (b) the optimal kernel parameters for the single-scale model with the Laplacian matrix are $\kappa = 19.5$ and $\tau = 2.5$ with a median Pearson's correlation coefficient $R_p = 0.795$.

3.3.3 CASF-2013 benchmark

Subsequently, we consider the CASF-2013 benchmark dataset from the PDBbind database. Since the CASF-2013 training set is smaller than the previously discussed CASF-2016 set, one might expect lower performance compared to the above results. However, the CASF-2013 benchmark will further confirm the robustness of the proposed model with less informative training data. Figure 3.5 captures the CV results of CASF-2013 benchmark dataset for the SYBYL atom-type model. The AGL-EAT model with the Adjacency matrix shows optimal kernel parameters $\kappa = 5.5$ and $\tau = 2.0$ with kernel type $\beta = \Phi_E$, resulting in a median Pearson's correlation co-

efficient $R_p = 0.795$. On the other hand, the model with the Laplacian matrix has optimal kernel parameters $\kappa = 4.5$ and $\tau = 2.0$ with kernel type $\beta = \Phi_E$, delivering a median $R_p = 0.796$.

Once the top-performing models for this benchmark are identified, we evaluate the performance of our model on the corresponding test set by calculating Pearson's correlation coefficient between predicted and experimental binding affinities. Following CASF-2016, a similar repetition of model training, totaling 50 cycles, both for the refined set and the general set is undertaken to generate predicted values, which are subsequently averaged to derive the final prediction. The performance summary of the top AGL-EAT models for the CASF-2013 benchmark is outlined in Table 3.4, with the AGL-EAT$^{\text{Adj}}_{\Phi_E,5.5,2.0}$ model achieving the highest Pearson correlation coefficient ($R_p = 0.845$). A visual comparison in the benchmark, depicted in Figure 3.4a, demonstrates our model's leading performance against competing models, underscoring its effectiveness in the evaluation.

Table 3.4: Performance of various AGL-EAT-Score models on CASF–2013 test set.

Model	Trained with Refined Set	Trained with General Set
AGL-EAT$^{\text{Adj}}_{\Phi_E,5.5,2.0}$	0.812	0.845
AGL-EAT$^{\text{Lap}}_{\Phi_E,4.5,2.0}$	0.804	0.841

3.3.4 CatS Dataset

In the context of the CatS dataset, we employ Kendall's tau correlation coefficient as the performance evaluation metric, which assesses the model's ability to capture the ranking and correlation of predicted binding affinities with the actual values, providing a comprehensive evaluation of the model's performance. The hyperparameter optimization process for this dataset follows a similar approach as used in the previous two benchmarks. Figure 3.6 displays the cross-validation results for the CatS dataset utilizing the SYBYL atom-type model. The optimal kernel parameters for the Adjacency matrix are $\kappa = 12.5$ and $\tau = 8.0$ with exponential kernel type producing a median Kendall's tau of 0.57837. The optimal kernel parameters with the Laplacian matrix are $\kappa = 16.5$ and $\tau = 10.0$ for $\beta = \Phi_E$ with a median Kendall's tau of 0.57305.

After having the best-optimized model for the CatS training set, we assess the performance on the test set by calculating Kendall's tau correlation coefficient between the predicted and experimental binding affinities. We train each of these optimized models using the training set and subsequently generate predictions for the test set. We repeat this process up to 50 times and calculate the average of all predicted values to yield the final predicted set, from which we calculated Kentall's tau correlation coefficient (Kendall's τ) and Spearman's rho correlation coefficient (Spearman's ρ). Table 3.5 reports the performance of our models for CatS dataset. The best-performing model for this dataset is the AGL-EAT$^{\text{L}}_{\Phi_E,5.5,2.0}$, achieving a Kendall's tau

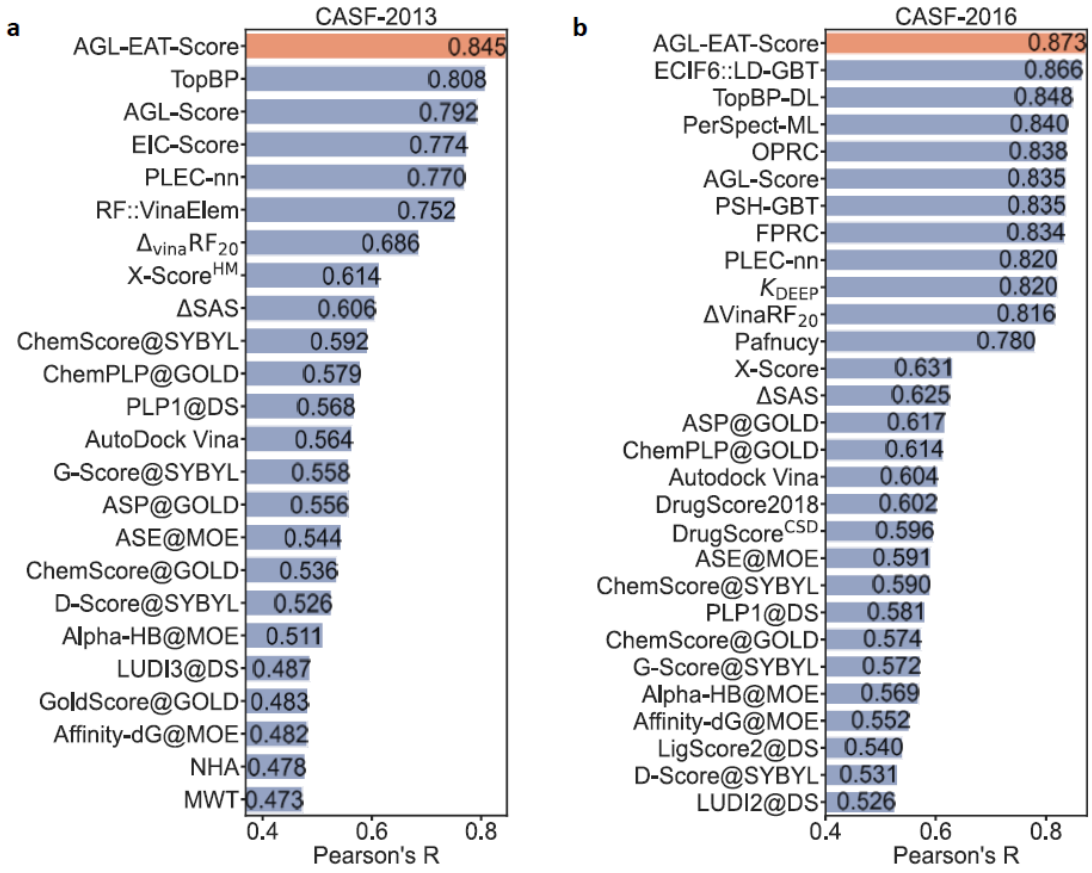

Figure 3.4: Performance comparison plot measured in Pearson's correlation of our AGL-EAT-Score and other machine learning-based models on the a) CASF-2013, and b) CASF-2016 benchmark datasets. Our model AGL-EAT-Score (highlighted in red color) scores. In CASF-2013 benchmark, our model achieved $R_p = 0.845$ and the results of other methods obtained from prior research [130, 76, 72]. In CASF-2016 benchmark, our model achieved $R_p = 0.873$ and the results of other methods obtained from prior research [118, 116, 130]

of 0.552, and a Spearman's ρ of 0.742. Figure 3.7 provides a performance comparison between our AGL-EAT-Score and other machine learning-based models taken from the official results of the D3R GC4 [98]. The results show that our model outperforms the affinity ranking of 459 CatS compounds in terms of Kendall's τ and Spearman's ρ.

3.4 Validation of the Robustness of AGL-EAT Model

3.4.1 Non-Redundant Training Sets

The performance of machine learning scoring functions is known to be influenced by the size of the training set and the degree of similarity between the training set and

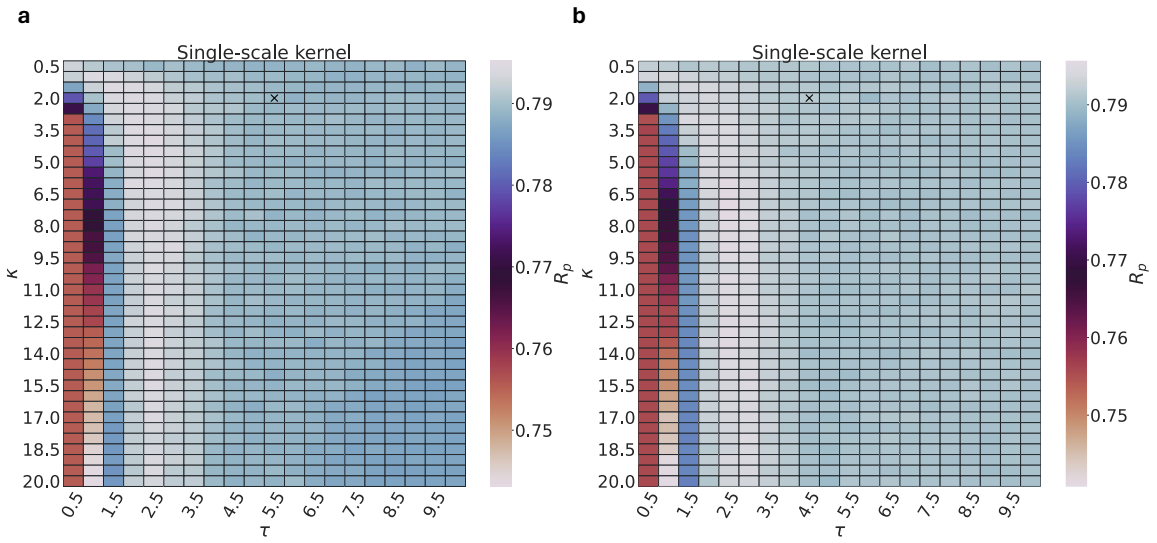

Figure 3.5: The optimized kernel parameters for the algebraic graph learning model with multiple atom types on the CASF-2013 dataset are indicated by 'x' marks, representing the best parameter. The optimal parameters for (a) single-scale model with the Adjacency matrix shows optimal kernel parameters: $\kappa = 5.5$ and $\tau = 2.0$, resulting in a median Pearson's correlation coefficient $R_p = 0.795$ and, (b) the single-scale model with the Laplacian matrix has optimal kernel parameters $\kappa = 4.5$ and $\tau = 2.0$, delivering a median $R_p = 0.796$.

Table 3.5: Performance of various AGL-EAT-Score models on CatS data set.

Model	Kendall's τ	Spearman's ρ
$\text{AGL-EAT}^{\text{Adj}}_{\Phi_E,5.5,2.0}$	0.539	0.729
$\text{AGL-EAT}^{\text{Lap}}_{\Phi_E,4.5,2.0}$	0.552	0.742

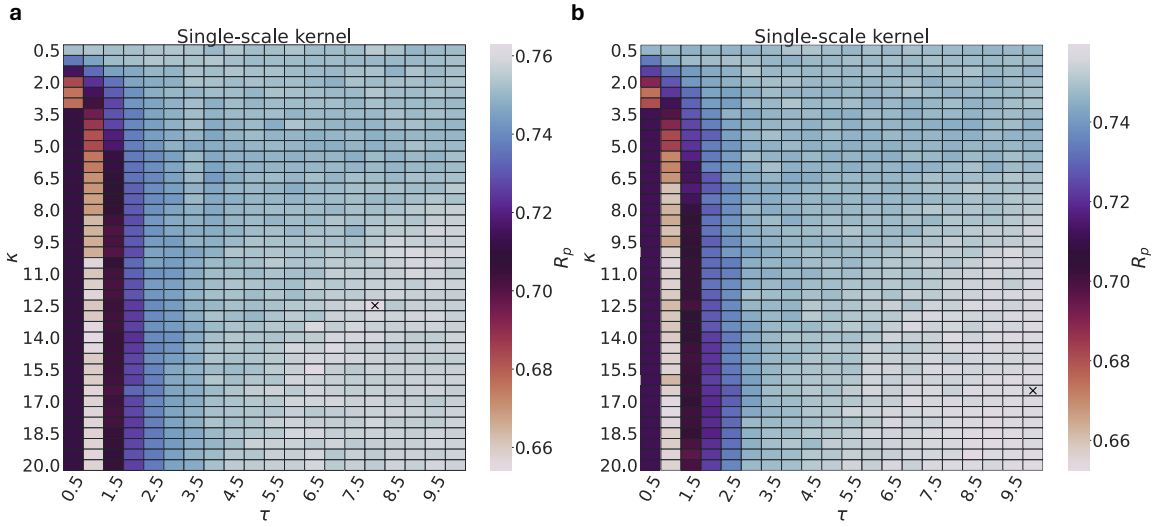

Figure 3.6: The optimized kernel parameters for the algebraic graph learning model with multiple atom types on the CatS dataset are indicated by 'x' marks, representing the best parameter. The optimal parameters for (a) single-scale model with the Adjacency matrix are $\kappa = 12.5$ and $\tau = 8.0$ with a median Kendall's $\tau = 0.57837$ and, (b) single-scale model with the Laplacian matrix are $\kappa = 16.5$ and $\tau = 10.0$ with a median Kendall's $\tau = 0.57305$

the test set. We investigated our model's performance on the multiple non-redundant training sets to address this issue. These non-redundant training sets are datasets that have been carefully curated to ensure that the included complexes are distinct from the test set complexes and do not contain redundant or highly similar complexes with test complexes. These sets are designed to reduce bias and overfitting in machine learning models by providing diverse and representative training data.

Similarity Computation

In our endeavor to construct a non-redundant training set, we explored the evaluation of three distinct similarity measurements between protein-ligand complexes. The first of these measurements, known as protein sequence similarity and denoted as P_s, quantifies the likeness between protein sequences. The second metric, L_s, deals with the structural resemblance of ligands. Lastly, the third measurement, BS_s, concerns itself with the comparison of protein-ligand binding sites. These similarity metrics collectively underpin the process of generating our non-redundant training set, allowing us to make informed decisions regarding redundancy reduction in the dataset. To compute the sequence similarity of two protein structures, we used "*ggsearch*36" from FASTA (version 36.3.8)[101], which employs a global-global (Needleman-Wunsch)

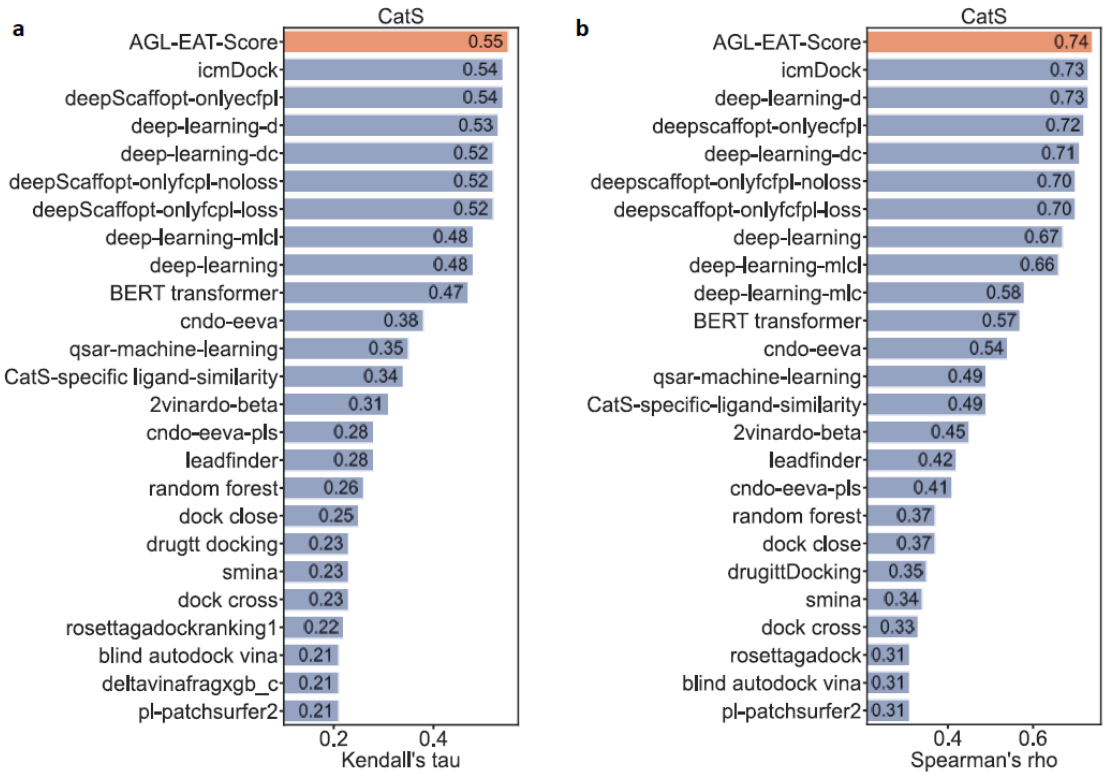

Figure 3.7: Performance comparison plot of our AGL-EAT-Score and other machine learning-based models on the Cathepsin S (CatS) dataset. Our model AGL-EAT-Score (highlighted in red color) scores Kendall's $\tau = 0.55$ and Spearman's $\rho = 0.74$.

search algorithm. The ligand structure similarity was computed using ROCS (version 3.5.1.1)[96, 52], which employs a Gaussian function with smoothness characteristics to model the molecular volume, allowing for systematic optimization to achieve the most accurate global fit. The binding site similarity was computed using PocketMatch (version 2.1) [138, 90], which assesses the similarity of binding sites using structural descriptors like residue properties and interatomic distances. This tool can also provide atomic-level alignments derived from pairings of amino acid residues.

To initiate the search for non-redundant complexes, we eliminated the overlaps between the train and test sets. We will be adopting the terms "hard overlap" to refer to the complexes that overlap between the train set and test sets, and "soft overlap" to refer to the structurally similar complexes in the train set and test sets, as defined by Minyi Su et al [117].

The process of searching for non-redundant training sets has been illustrated in the following contexts and visually represented in Figure 3.8.

1. Training Set vs Test Set Similarity Evaluation: We start by eliminating any "hard overlap" complexes shared between the training set and the test set. We then proceed to evaluate the similarity between every complex in the training set against every complex in the test set by employing the three similarity metrics, P_s, L_s & BS_s

as discussed above.

2. Training Set vs Test Set Redundancy Reduction: A training complex was classified as redundant to the test set if all three similarity metrics were above the defined cutoff and further eliminated from the training set.

3. Training Set Internal Similarity Evaluation: Following that, we compute the similarities among the remaining complexes within the training set. If the measurements for all three similarity metrics between two complexes exceed the defined cutoff, we categorize these complexes as redundant to each other.

4. Training Set Internal Redundancy Reduction: Finally, we adopt a systematic approach to eliminate redundant samples from the training set to get the optimal training sets for different similarity cutoffs.

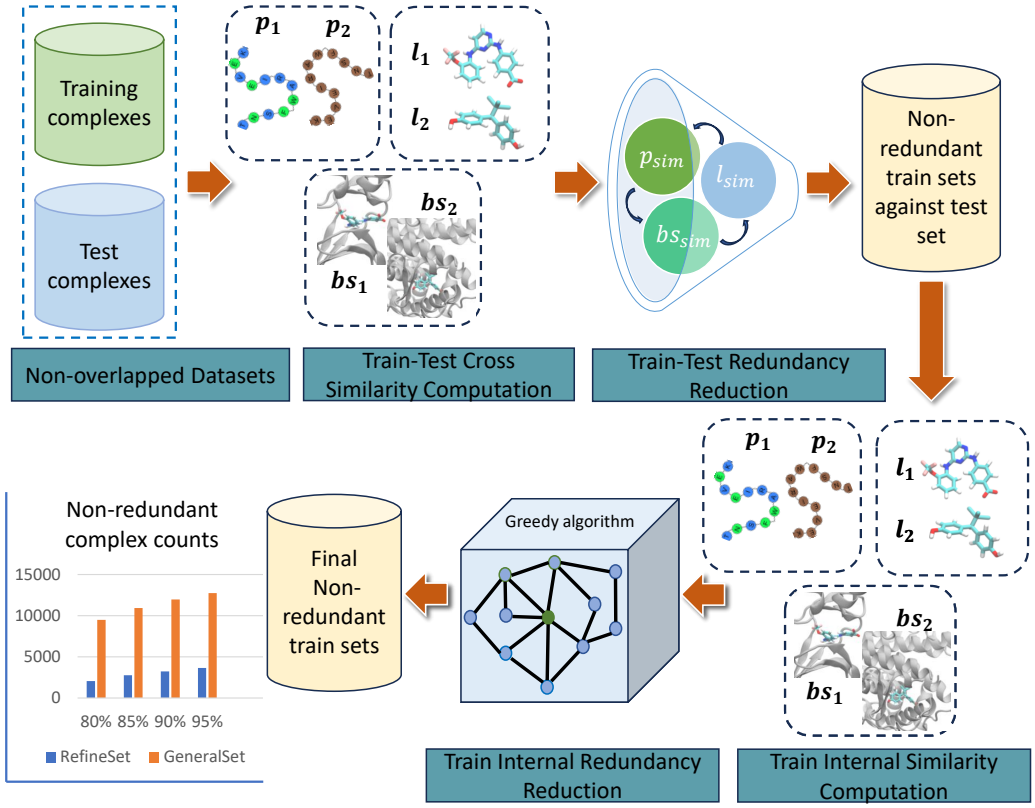

Figure 3.8: Visualization of the strategy for searching a non-redundant training set for different similarity cutoffs. In the initial phase, we conducted calculations to determine the cross-similarity between the training and test sets, followed by a process to reduce redundancy. In the following phase, we performed internal similarity calculations on the training set and subsequently reduced redundancy to get the final non-redundant training sets.

PDBbind v2016: In our study, we explored the calculation of similarity for both the PDBbind v2016 general set and the refined set. We derived non-redundant training sets for six distinct similarity cutoff points: 70%, 75%, 80%, 85%, 90%, and 95%.

The summary of non-redundant complexes for different similarity cutoffs is captured in Figure 3.9.

We initiated the process by addressing the similarity evaluation between the training set, first considering the refined set, and the test set, which is the core set. First, we identified and eliminated any "hard overlap" complexes, which are those present in both the refined set and the core set. This step ensured that there were no identical complexes shared between the two sets.

Following the removal of hard overlaps, we evaluated the similarity between each complex in the refined set and every complex in the core set using three similarity metrics: Protein Sequence Similarity (P_s), Ligand Similarity (L_s), and Binding Site Similarity (BS_s).

A training complex was classified as redundant if all three similarity metrics (P_s, L_s, and BS_s) exceeded the defined similarity cutoff when compared to any complex in the test set. These redundant complexes were then removed from the refined set to maintain its non-redundant nature and ensure that the model is trained on unique data, preventing data leakage and overfitting.

After addressing the redundancies relative to the core set, we proceeded to compute the internal similarities among the remaining complexes within the refined set. For each pair of complexes in the refined set, if the similarity scores for all three metrics (P_s, L_s, and BS_s) exceeded the cutoff, they were marked as redundant to each other. This step was crucial for identifying and eliminating internal redundancies within the training set, ensuring that the training data was diverse and non-redundant.

To systematically eliminate redundant samples and obtain the optimal training sets for different similarity cutoffs, we employed a systematic approach. This approach involved iteratively removing redundant complexes until the training set met the desired criteria for each similarity cutoff point (70%, 75%, 80%, 85%, 90%, and 95%). We implemented the same methodology for the PDBbind v2016 general set as well to obtain the non-redundant training sets.

However, it's important to note that a similar comparison for the PDBbind v2015 dataset wasn't conducted since nearly all its molecules are already included in the PDBbind v2016 dataset, with only a marginal difference of around 10%.

CatS: When working with the CatS dataset, we decided to explore the redundant complexes, where multiple complexes share common features or functions, rather than focusing on non-redundant complexes. Our approach to identifying redundant complexes in the CatS dataset closely paralleled the methodology we employed for investigating non-redundant complexes in the PDBbind dataset.

To begin with, we derived redundant training sets for 10 distinct similarity cutoffs, ranging from 45% to 90%. This process involved computing the similarities between complexes in the dataset using three key similarity metrics: Protein Sequence Similarity (P_s), Ligand Similarity (L_s), and Binding Site Similarity (BS_s). These metrics were crucial for assessing the degree of redundancy among the complexes.

For each similarity cutoff point, we evaluated the similarity between every pair of complexes within the CatS dataset. Complexes were considered redundant if the similarity scores for all three metrics (P_s, L_s, and BS_s) exceeded the defined cutoff. This approach ensured that the identified redundant complexes shared significant common features or functions.

Following the identification of redundant complexes, we created redundant training sets for each similarity cutoff. This allowed us to systematically explore how different levels of data redundancy impacted the model's performance. By training our proposed model on these redundant datasets, we aimed to understand how well the model generalizes and performs across varying degrees of data redundancy.

Redundant complexes often share similar features, which can be advantageous for model training. By focusing on these shared features, we can refine and optimize the training set, emphasizing the importance of critical features. As a result, the model is better equipped to recognize and utilize these features during training, potentially leading to improved performance. The summary of redundant complexes for different similarity cutoffs for the CatS dataset is presented in Figure 3.10.

Performances on PDBbind v2016 Non-redundant Training Sets

Indeed, it is widely recognized that the size of the training set and the degree of similarity between the training set and the test set have a profound impact on machine learning scoring functions. To this end, we aim to investigate our model's performance on the multiple non-redundant training sets of PDBbind v2016 General set and Refined set with different levels of redundancy.

Figure 3.9 visualizes the performance of our proposed model when trained on these non-redundant training sets. Interestingly, when we calibrate our model on a training set that shares a high level of similarity with the test set, for example, employing a 95% similarity cutoff from the General set, we achieved a Pearson correlation of 0.869. This is remarkably close to the Pearson correlation of 0.873 obtained when using the complete training set. These findings emphasize the consistent and robust nature of our model across varying non-redundant training sets, without experiencing significant drops in predictive capabilities. This robust performance underscores the model's reliability and versatility in handling diverse datasets.

Investigating the performances across various non-redundant training sets reveals another interesting fact: data quality significantly influences the model's effectiveness. Consider the refined non-redundant training set for the 95% similarity cutoff, notably smaller in size compared to the general non-redundant training set at the 70% similarity cutoff. However, the performance of the smaller refined non-redundant set substantially outperforms the larger set. This observation underscores the importance of data quality over quantity, highlighting how the focused, refined data yield more accurate models despite their smaller scale.

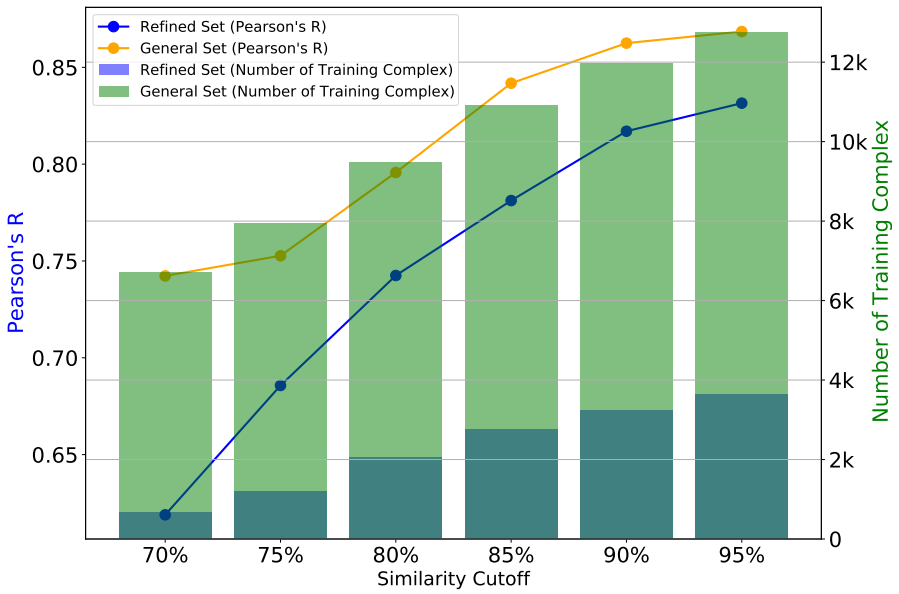

Figure 3.9: The performance comparison plot illustrates the non-redundant training sets of PDBbind v2016 RefinedSet and GeneralSet. Green bars indicate the count of non-redundant training complexes for the PDBbind v2016 General Set across different similarity cutoffs, while blue bars represent the count for the PDBbind v2016 Refined Set. Additionally, orange lines show the Pearson correlation coefficient (R_p) for the General Set, and blue lines depict R_p for the Refined Set, both for various non-redundant training sets across various similarity cutoffs.

Performances on CatS redundant Training Sets

In exploring the CatS dataset's redundant training complexes, we observed a nuanced impact of redundancy levels on our model's performance. Training the model on various training sets with varying similarity cutoffs, we noted that higher redundancy maintained stable performance, as measured by Kendall's Tau. Figure 3.10 illustrates the counts of different redundant training sets for similarity cutoff ranging from $45\% - 90\%$ and the performances of the corresponding redundant training sets. As captured in Figure 3.10, for 45% similarity cutoff we achieved a Kendall's Tau = 0.5512, which remains quite stable up to 75% similarity cutoff with Kendall's Tau = 0.5355. This stability suggests the model effectively leverages redundant training complexes without significant loss in predictive power. However, beyond this threshold, performance declined, highlighting a balance between leveraging redundancy for rich feature extraction and the diminishing returns of too-similar training data, underscoring the model's adaptability within a spectrum of redundancy.

3.4.2 Importance of Extended Atom Types

We further performed an investigation by expanding our previous work, AGL-Score[93] to justify the impact of the protein-ligand extended atom type (EAT) features in the

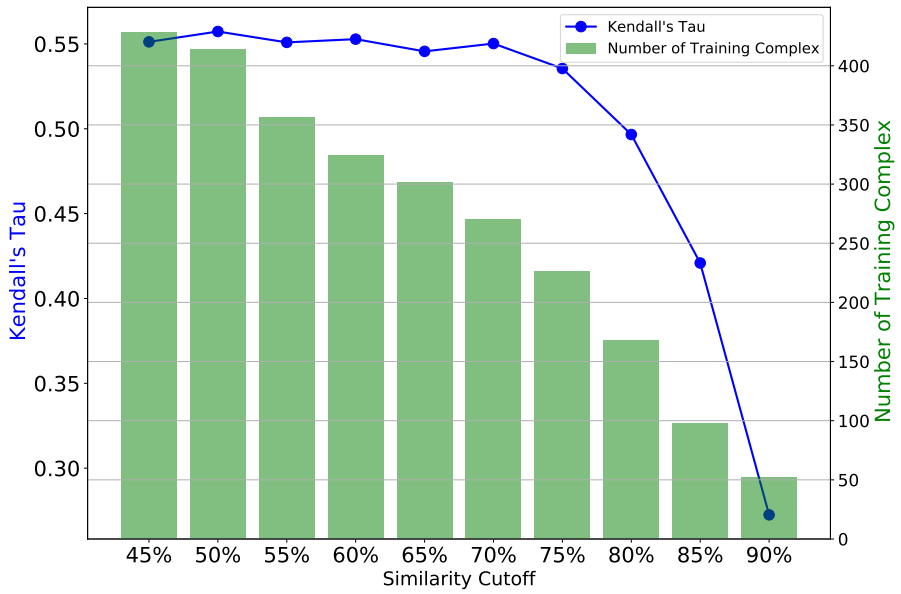

Figure 3.10: The performance comparison plot of redundant training sets of CatS Dataset. Green bars indicate the count of redundant training complexes for the CatS across different similarity cutoffs, and the blue lines depict Kendall's τ for the CatS redundant datasets across various similarity cutoffs.

model's performance. We follow a systematic approach throughout this investigation.

We initiated the process by ranking the important features of both the AGL-EAT-Score and AGL-Score models, using the PDBbind v2016 general set features as a reference. Figure 3.11 illustrates the top ten important features of the AGL-EAT-Score and AGL-Score models in this dataset.

As shown in Figure 3.11a, some of the most important interactions of AGL-Score are C-C, O-C, N-C so on. For the first set of experiments, we replaced the C-C features in AGL-Score with the extended atom type features from AGL-EAT-Score, specifically the C-C.1, CA-C.1, CB-C.1, C-C.2, CA-C.2, CB-C.2, and so on. We then employed this modified feature set to predict the test dataset. To ensure the reliability of predictions, we repeated the process 50 times and obtained the final predicted set by averaging all the predicted values. In the case of the base AGL-Score model, we reported a performance of $R_p = 0.8559$ on the PDBbind v2016 general set and evidently, incorporating the C-C extended atom type (EAT) features led to an improvement in performance, resulting in $R_p = 0.8685$.

We also carried out similar analogous feature replacements for O-C, N-C, C-O, and O-H interactions, utilizing their extended atom-type features, and the modified feature sets were used for test dataset predictions. A performance comparison of these experiments is presented in Figure 3.12. In our final set of experiments, we replaced all the base AGL-Score C-C, O-C, N-C, and O-H interaction features with the corresponding AGL-EAT-Score extended atom type features for prediction purposes, resulting in improved performance with an $R_p = 0.8715$, named as AGL-Score-

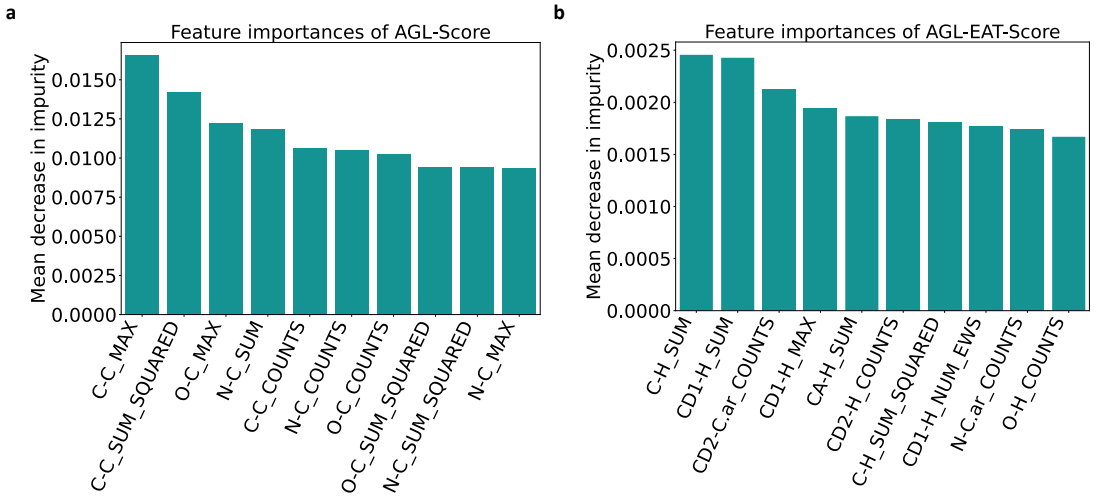

Figure 3.11: Feature importance of (a) AGL-Score [93] (b) our AGL-EAT-Score model on PDBbind v2016 general set using the mean decrease in impurity of the gradient boosting trees. The horizontal axis on the graph denotes the names of the features, which are represented as various statistical metrics (like sum, mean, median, and others) specific to the atom name groups in the ligand-protein interaction.

combined-eat features in Figure 3.12, indicates a significant enhancement. These experiments validate that the detailed consideration of atom types contributes positively to predictive accuracy, highlighting the importance of nuanced feature representation in computational modeling.

3.5 Conclusion

In this work, we propose a novel model named Algebraic Graph Learning with Extended Atom-Type Scoring Function (AGL-EAT-Score), which exhibits high accuracy in capturing protein-ligand interaction information. This model is distinguished by its innovative integration of extended atom-type multiscale weighted colored subgraphs and algebraic graph learning, enabling a detailed and sophisticated representation of molecular interactions. The AGL-EAT-Score has demonstrated its efficacy in accurately predicting ligand-receptor binding affinities, showcasing superior performance compared to both traditional and contemporary machine learning-based scoring functions. This was evidenced through extensive evaluations using benchmark datasets such as CASF-2016, CASF-2013, and the CatS dataset. The computational efficiency of our model, which requires only atom types and coordinates as input, further enhances its practicality. The AGL-EAT features for a given protein-ligand complex

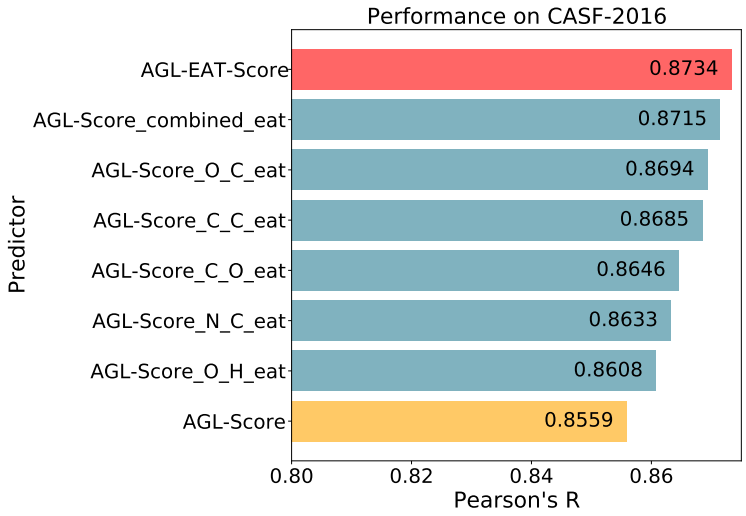

Figure 3.12: Performance comparison plot of base AGL-Score [93] (highlighted in yellow color), AGL-EAT-Score (highlighted in red color), and the modified AGL-Score with different extended atom type interaction features from AGL-EAT-Scre (highlighted in blue colors) on the PDBbind v2016 General set.

can be calculated in under a second on a standard CPU, making it an ideal tool for large-scale screenings and real-time applications.

To further validate the robustness and address concerns of overfitting in machine learning-based scoring functions, we tested the performance of the proposed AGL-EAT-Score against redundant and non-redundant data built on the PDBbind general set v2016 and the CatS dataset. The model's performance, consistent with the level of training data information, confirms the necessity of incorporating extended atom-type information rather than relying solely on basic element types.

As the field of drug design continues to progress, the proposed AGL-EAT-Score is positioned as a robust, innovative, and essential tool for describing the complex landscape of molecular interactions, thereby contributing significantly to advancements in pharmaceutical research.

4.1 Backgrounds

4.1.1 Blood-Brain Barrier Permeability

The blood-brain barrier (BBB) is a selective permeability barrier that separates the circulating blood from the brain extracellular fluid in the central nervous system (CNS) [27]. The BBB serves a crucial role in maintaining the homeostasis of the brain's microenvironment, protecting the brain from potentially harmful substances and pathogens while allowing essential nutrients and gases to pass through [64]. This barrier is composed of endothelial cells that line the brain capillaries, astrocyte end-feet, and pericytes, all of which contribute to its selective permeability [143].

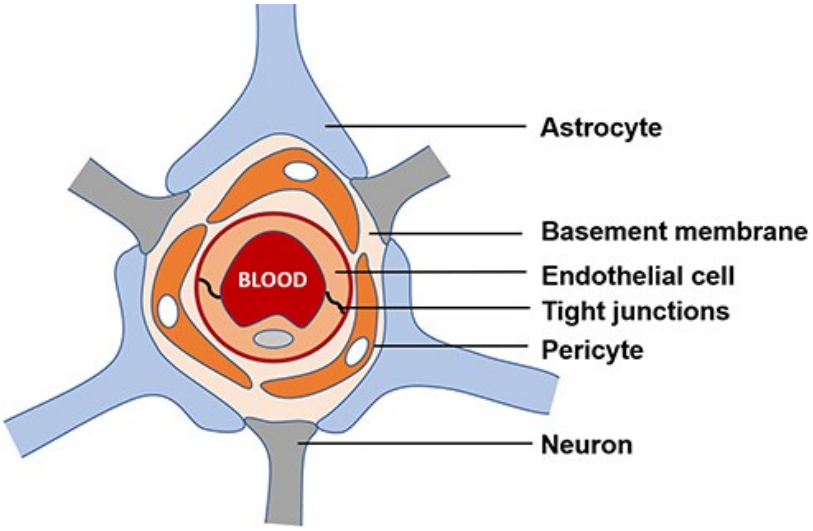

Figure 4.1: An illustration of blood-brain Barrier structure [Image: Zhang et al. (2021)]

BBB permeability (BBBP) refers to the ability of substances to cross the barrier from the blood into the brain tissue. This permeability is severely restricted by tight junctions and adhesion junctions in the endothelial cells of the BBB, which prevent large and hydrophilic molecules from passing through. Only small, lipophilic molecules or those with specific transport mechanisms can typically cross the BBB [64].

Understanding and predicting BBB permeability is essential for drug development, particularly for treatments targeting CNS disorders. The ability to cross the BBB is a critical factor in determining the potential efficacy and safety of a compound. Drugs that cannot effectively cross the BBB may fail to reach therapeutic concentrations in the brain, limiting their effectiveness in treating CNS diseases. Conversely, compounds that cross the BBB too readily may cause adverse effects by disrupting

the delicate balance of the brain's microenvironment. The ineffective transportation of molecules across the BBB poses a significant challenge in drug design for CNS diseases [128, 46, 32]. Various experimental and computational approaches have been employed to predict BBB permeability, leveraging the properties of molecules and advanced modeling techniques.

In recent years, various in vivo and in vitro models have been established to evaluate BBB permeability [30, 1, 18, 85]. In vivo models, which involve animal studies, provide physiologically relevant insights by observing compound transport within living organisms. These models are considered the gold standard for BBBP assessment due to their ability to mimic the complex interactions in a living system [7]. However, ethical concerns, high costs, and time-consuming protocols limit their widespread use. In vitro models, on the other hand, use cultured brain endothelial cells to replicate the BBB, allowing for controlled testing environments. These models offer a balance between physiological relevance and experimental control, making them a valuable tool in early-stage drug development. Techniques such as dynamic models [24, 25] and co-culture models [92, 91] have improved the accuracy and throughput of in vitro BBB studies. Despite their reliability, these methods are often expensive and time-consuming.

To mitigate these challenges, in silico approaches have emerged, using computational techniques to predict BBB permeability efficiently and cost-effectively by leveraging molecular properties and advanced algorithms. These computational models can screen large libraries of compounds quickly, identifying potential BBB-permeable drugs before experimental testing. In silico methods also allow for the exploration of molecular modifications to improve BBB permeability. These in silico approaches can be categorized into quantitative and qualitative models, each serving distinct purposes in the prediction process. Quantitative BBBP prediction models provide a more detailed assessment by predicting numerical values that represent the extent of permeability. These models are crucial for drug development, offering precise information about how much of a compound can cross the BBB. Quantitative structure-activity relationship (QSAR) models are widely used, employing statistical techniques [142, 139, 26, 69, 81] to relate molecular descriptors to permeability values. The goal of these quantitative methods is to predict specific properties describing the BBB permeability of compounds, such as logBB (the logarithm of the brain-to-blood concentration ratio) and logPS (the permeability-surface area product) [3, 97]. On the other hand, qualitative BBBP prediction models classify compounds into broad categories, such as BBB+(high permeability) and BBB- (low permeability). These models rely on identifying patterns and properties in molecular structures that indicate their ability to cross the BBB. Commonly used Machine learning classifiers, like support vector machines (SVMs) [75, 54], genetic algorithms [113] and decision trees [144], are trained on datasets of known compounds to learn which molecular descriptors are indicative of high or low permeability.

Additionally, advanced computational techniques, such as deep learning models [119, 86, 132], have shown promise in enhancing BBBP predictions. DL models such as convolutional neural networks (CNNs), graph neural networks (GNNs), and recurrent neural networks (RNNs) can learn complex, non-linear relationships between

molecular features and leverage large datasets to improve prediction accuracy significantly.

4.2 Methods and Materials

In this section, we discuss the comprehensive development of the geometric graph learning (GGL) score model applied for predicting blood-brain barrier permeability (BBBP). Additionally, we demonstrate that the geometric graph learning (GGL) framework can be effectively combined with various machine learning and deep learning models to predict target properties, including random forests, support vector machines (SVMs), convolutional neural networks (CNNs), and graph convolutional Neural Networks (GCNNs) [136].

4.2.1 Geometric Graph Learning

Geometric Graph Learning Ligand-only Weighted Colored Subgraphs Features for Extended Atom Type: Our GGL ligand-only model employs a sophisticated geometric graph learning framework, incorporating extended atom-type features to predict ligand binding affinity. This approach integrates the structural and chemical properties of the ligand into weighted colored graphs and coupled with different machine learning and deep learning models for superior predictive accuracy. The detailed methodology of GGL extended atom-types ligand-only model is described in Section (2.4). Here, we briefly present the development of the model.

The model begins by constructing a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ representation of the ligand molecule, where $\mathcal{V}$ is the set of vertices and $\mathcal{E}$ is the set of edges. Each atom in the ligand is treated as a node, and the interactions or bonds between these atoms are represented as edges. The graph is then enhanced through the process of graph coloring, wherein different types of atoms are assigned distinct labels. This coloring process helps encode various chemical properties and interaction types into the graph structure. Following our AGL-EAT-Score model, we have employed the SYBYL atom types for the ligands to capture the detailed chemical description. The use of SYBYL atom types allows for precise classification, differentiating between various subtypes of all atoms.

We use commonly selected forms for the subgraph weight function, specifically the generalized exponential function or the generalized Lorentz function (Equation (2.9)). These functions provide flexible and robust methods for evaluating interaction strengths, ensuring that they appropriately diminish with increasing distance. This approach captures the essential characteristics of non-covalent interactions within molecular structures.

Analyzing these weighted colored subgraphs allows us to extract detailed molecular descriptors and explore the system's multiscale behavior. This involves considering various characteristic distances for different atom type pairs, enabling the creation of diverse and scalable graph-based descriptors. One such descriptor is the geometric subgraph centrality (GSC), which sums the contributions from all subgraph edges.

This comprehensive method facilitates accurate modeling and analysis of molecular behavior by capturing intricate molecular interactions and properties.

Geometric Graph Learning Ligand-only Atom Features: To generate the GGL atom features to integrate with Chemprop atom-level features, we followed a simple architecture. We start with constructing a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ representation of the ligand molecules, where $\mathcal{V}$ is the set of vertices representing the atoms, and $\mathcal{E}$ is the set of edges representing the interactions between the atoms. The $\mathcal{V}$ denoted as

$$\mathcal{V} = \{r_1, r_2, \cdots, r_N\} \tag{4.1}$$

with N atoms in the graph $\mathcal{G}$. The non-covalent interactions between any pair of atoms can be expressed using rapidly decreasing weight function

$$\mathcal{E} = \{\Phi(\|\mathbf{r}_i - \mathbf{r}_j\|; \eta_{ij}) | i \neq j; i, j = 1, 2, \cdots, N; \ \|\mathbf{r}_i - \mathbf{r}_j\| \leq d\}, \tag{4.2}$$

where $\|\mathbf{r}_i - \mathbf{r}_j\|$ denotes the Euclidean distance between the ith and jth atoms of $\mathcal{G}$. The weight function Φ assesses the interaction strength between atoms, taking into account their Euclidean distances. The weight function Φ satisfies specific boundary conditions: it approaches 1 as the distance between two atoms approaches zero and approaches 0 as the distance becomes very large. These conditions ensure that the interaction strength decreases rapidly with increasing distance, accurately reflecting the nature of non-covalent interactions.

For our purposes, we selected Φ as the generalized exponential function, denoted as follows:

$$\Phi(\|\mathbf{r}_i - \mathbf{r}_j\|; \eta_{ij}) = e^{-(\|\mathbf{r}_i - \mathbf{r}_j\|/\eta_{ij})^\kappa}, \quad \kappa > 0, \tag{4.3}$$

where κ is a parameter that controls the rate of decay, and η_{ij} is a scaling factor for the distance between ith and jth atoms and defined as $\eta_{ij} = \tau(r_i + r_j)$, where r_i and r_j are the van der Walls radii for ith and jth atoms.

$$\mu(\eta_{ij}) = \sum_{i \neq j} \mu_i(\eta_{ij}) = \sum_i \sum_{j \neq i} \Phi(\|\mathbf{r}_i - \mathbf{r}_j\|; \eta_{ij}), \tag{4.4}$$

where $\mu(\eta_{ij})$ is the geometric centrality for ith atom to all other atoms.

To analyze the statistical descriptors generated from the ligand graph in either way described above, we employed machine learning and deep learning algorithms. Our approach leverages supervised machine learning algorithms for both classification and regression tasks. In this context, the labeled dataset is divided into two subsets: a training set and a test set. The optimization process for training a machine learning model can be reformulated as the following loss minimization problem for a function $\mathcal{G}(\mathcal{X}_i, \lambda)$ that encodes the geometric information of a molecule into suitable graph representations:

$$\min_{\lambda, \theta} \sum_{i \in I} \mathcal{L}(\mathbf{y}_i, \mathcal{G}(\mathcal{X}_i, \lambda); \theta) \tag{4.5}$$

In this equation, where $\mathcal{X}_i$ is a labeled dataset corresponding to the ith data point in the training set, and λ is a set of kernel parameters, $\mathcal{L}$ denotes a scalar loss function that needs to be minimized, y_i represents the labels assigned to the ith sample in the training set I , and θ includes hyperparameters specific to the chosen machine learning algorithm, which are typically tuned to achieve optimal performance.

The objective of this optimization is to find the best set of kernel parameters and hyperparameters that minimize the loss function, thereby enhancing the model's accuracy and predictive performance on unseen data.

We utilize several commonly used machine learning and deep learning frameworks to establish baseline accuracies and demonstrate the overall advantage of GGL-score in our prediction tasks. These frameworks include random forests (RF), support vector machines (SVMs), convolutional neural networks (CNNs), and Directed Message Passing Neural Network (DMPNN). We have also examined our model's performance using CatBoost and XGBoost on the $BBBP_{d_2}$ dataset.

4.2.2 GGL-Chem Fusion Graph Model

The main idea of this model is to integrate geometric subgraph features of 3D molecular structures captured by GGL with the 2D molecular graph features of DMPNN to enhance the prediction of blood-brain barrier permeability (BBBP).

Here, we explicitly discuss the methodology of the GGL-Score model being coupled with Chemprop [53], a versatile machine learning software package designed for the prediction of molecular properties. It leverages Directed Message Passing Neural Networks (D-MPNNs) [136], a class of graph-convolutional neural networks, to learn atomic and molecular embeddings from molecular graphs.

GGL focuses on capturing the spatial and topological features of 3D molecular structures. By representing molecules as graphs where atoms are nodes and bonds are edges, GGL effectively analyze the geometry of these structures. The spatial arrangement of atoms and the distances between them are critical in understanding molecular interactions and properties. In the GGL model, molecules are decomposed into colored subgraphs, with each color representing a specific atom type or group of atom types. This decomposition allows the model to consider the distances between atoms within these subgraphs, providing detailed spatial information.

Whereas DMPNNs, as implemented in Chemprop, excel at learning complex molecular interactions and properties from 2D molecular graphs. These networks pass messages along the edges of the graph, enabling the aggregation of information from neighboring atoms to each node. This directed message-passing mechanism allows the model to capture the directionality and dependencies of chemical bonds, which are crucial for accurate property prediction.

By integrating GGL features with DMPNN, the model significantly enhances the understanding of how molecular features influence BBB permeability, thereby improving prediction accuracy. In the next section, we will begin by providing an overview of the Chemprop [53] method. Following this, we will introduce the GGL-Chem fusion model, which integrates geometric graph learning with D-MPNN for enhanced pre-

diction capabilities. Key model configurations and settings used in our experiments will also be detailed.

4.2.3 Chemprop

The Chemprop model is a sophisticated machine learning tool designed for predicting molecular properties. It employs Directed Message Passing Neural Networks (D-MPNNs) to process molecular graphs and extract meaningful features. Initially, molecular structures represented as SMILES strings are converted into graphs using RDKit [68], where atoms and bonds are encoded with specific features.

In the D-MPNN framework, directed edges are used to pass messages between nodes, updating hidden states iteratively and aggregating these into atomic embeddings. These embeddings are then combined into a single molecular embedding through various aggregation techniques, such as summation or averaging.

The molecular embeddings are further processed by a feed-forward neural network (FFN) to predict target properties. The FFN is customizable, with multiple layers and activation functions to refine the prediction process.

Directed Message Passing Neural Networks (DMPNN)

The Directed Message Passing Neural Network (DMPNN) is a specialized form of a Message Passing Neural Network (MPNN) was proposed by Yang et al [136]. Unlike traditional MPNNs, which use atom-centered messages, DMPNNs use bond-centered (directed edge) messages to propagate information through the molecular graph. This design aims to prevent the creation of redundant loops and reduce noise in the message passing process.

Each directed edge vw (representing a bond from atom v to atom w is initialized with hidden states h^0_{vw}. This is done by concatenating the features of the atoms and bonds, followed by a linear transformation and activation function.

$$h^0_{vw} = \tau(W_i \cdot \text{cat}(x_v, e_{vw})) \tag{4.6}$$

where τ is the ReLU activation function, $W_i \in \mathbb{R}^{h \times h}$ is a learned weight matrix, x_v are the features of atom v, and e_{vw} are the features of bond vw, which than concatenated by $\text{cat}(x_v, e_{vw}) \in \mathbb{R}^h$.

For each time step t the hidden states h^t_{vw} and messages m^t_{vw} are updated based on the incoming messages from neighboring edges.

$$m^{t+1}_{vw} = \sum_{k \in N(v) \setminus w} h^t_{vk} \tag{4.7}$$

where m^{t+1}_{vw} represents the message passed along the directed edge vw at the next time step. This message is computed as the sum of the hidden states h^t_{vk} from the neighboring edges k of node v, excluding the edge vw to prevent self-loops. This ensures that the message incorporates information from all relevant neighboring bonds.

The hidden state h_{vw}^{t+1} is then updated using the initial hidden state and the new messages:

$$h_{vw}^{t+1} = \tau(W_m \cdot h_{vw}^0 + W_u \cdot m_{vw}^{t+1}) \tag{4.8}$$

In this equation, the hidden state h_{vw}^{t+1} is updated by combining the initial hidden state h_{vw}^0 and the newly computed message m_{vw}^{t+1}. The terms W_m and W_u are learnable weight matrices that transform these inputs. The ReLU activation function τ is applied to introduce non-linearity and enable the model to learn complex relationships.

After T message passing steps, the final node representations are obtained by aggregating the hidden states of incoming edges. For each node v, the hidden state is aggregated as:

$$m_v = \sum_{w \in N(v)} h_{wv}^{(T)} \tag{4.9}$$

$$h_v = \tau\left(W_a \cdot \mathrm{cat}\left(x_v, m_v\right)\right) \tag{4.10}$$

This equation describes the process of aggregating information from the incoming edges to compute the final hidden state for each node v.

The molecular representation h_G is obtained by summing the final hidden states h_v of all nodes v in the graph:

$$h_G = \sum_{v \in V} h_v \tag{4.11}$$

This aggregation step produces a single vector that represents the entire molecule, capturing information from all atoms and bonds.

The molecular representation h_G is then used for property prediction through a feed-forward neural network:

$$\hat{y} = f(h_G) \tag{4.12}$$

In this final step, the molecular representation h_g is fed into a feed-forward neural network, which outputs the predicted property $\hat{y}$. This network can consist of multiple layers and activation functions designed to map the high-dimensional representation h_G to the target property.

In summary, the DMPNN architecture uses directed edge messages to propagate information through a molecular graph, initializing edge hidden states with concatenated atom and bond features, updating these states through message passing, aggregating node representations, and finally predicting molecular properties with a feed-forward neural network. This method reduces redundant loops and noise, enhancing the accuracy and robustness of molecular property predictions. The DMPNN algorithm is implemented using the Chemprop software package[53], which is available as an open-source tool.

To this end, two systematic approaches have been followed to combine the generated geometric subgraph features with the DMPNN graph features: atom-level feature combination and molecule-level feature combination. In the atom-level feature combination, the geometric graph features of individual atoms from the molecule are combined with their corresponding atom features from the molecular input graphs of the DMPNN. This method allows for a detailed representation at the atomic level, capturing intricate relationships and interactions. Figure 4.2 illustrates GGL-Chem fusion model for atom-level features.

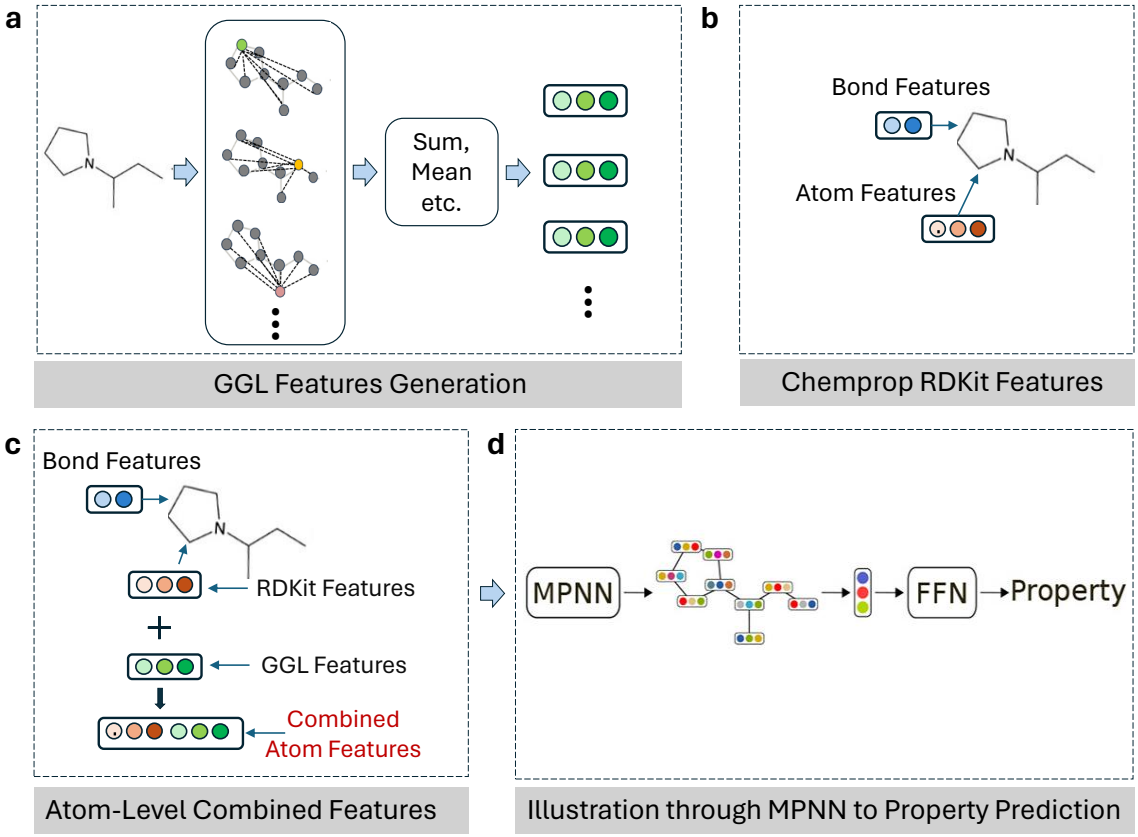

Figure 4.2: An illustration of GGL-Chem atom-level fusion graph model. a) Construct geometric graph learning atom features by considering statistical information (sum, mean, median, etc.) about the rigidity of the molecular graphs. b) Chemprop converts molecular SMILES string to molecular graph using RDKit, where atoms represent vertices and bonds edges, c) integrating GGL features to Chemprop' RDKit features as atom-level features d) Pass these combined features through a message passing neural network to update all feature vectors, followed by an aggregation function and a feed-forward neural network for property prediction.

In the molecule-level feature combination, the geometric features are aggregated at the molecular level and then combined with the features processed by the DMPNN for

the entire molecule. This approach provides a comprehensive view of the molecule's overall properties, enabling a broader analysis of the molecular structure as a whole.

4.2.4 Model Settings and Trainings

In this section, we detail the parameters set up for the training of various machine learning and deep learning models used in this study.

GGL-Chem: Chemprop take SMILES strings as input structures processed by RDKit [68]. The input features for atoms and bonds are consistent with those employed by Yang et al[136]. Chemprop's default standard parameter settings have been utilized to train the models, which include 30 epochs, the ReLU activation function, and the Adam optimizer, the batch size is usually set to 50. Additionally, the default hidden size for the feed-forward neural network (FFN) layers is set to 300, with a dropout rate of 0.1 and a weight decay of 0.01.

GGl$_{rf}$: We employ the RandomForestClassifier from scikit-learn v0.24.1, fine-tuned with specific parameters to optimize performance: 3800 estimators, a minimum of 10 samples required to split an internal node, a minimum of 2 samples required to be at a leaf node, 'auto' selection for the number of features to consider when looking for the best split, a maximum depth of 40, and enabling bootstrap sampling. These major hyperparameters are optimized using the grid-search method.

GGL$_{CNN}$: The Convolutional Neural Network (CNN) model used in this study is designed with several key components to effectively capture and process image data for binary outcome prediction. The model begins with three convolutional layers, each with an increasing number of filters (32, 64, and 128) and a consistent 4x4 kernel size. These layers utilize ReLU activation functions and a stride of 1, allowing the network to learn complex features from the input images. Following the convolutional layers, a flatten layer is employed to convert the 2D feature maps into a 1D vector, preparing the data for the dense layers. The model then incorporates two fully connected (dense) layers with 100 and 50 neurons, respectively. These layers are enhanced with L2 regularization set to 0.01 to prevent overfitting, ReLU activation for non-linear transformations, batch normalization to stabilize and accelerate the training process, and dropout for additional regularization. The output layer consists of a single neuron with sigmoid activation, specifically designed for binary outcome prediction. To optimize the model, we utilize Stochastic Gradient Descent (SGD) as the optimizer, with a specified learning rate, momentum of 0.9, a decay rate of 1e-6, and a clip value to prevent gradient explosion. Finally, the binary cross-entropy loss function is employed to measure the performance of the model, guiding the optimization process by minimizing the error in the predicted outcomes.

GGL$_{SVM}$: The SVM training parameter C is crucial in balancing model complexity and training error [140]. When C is set too high, the model risks overfitting as it attempts to minimize the penalty for misclassified points by incorporating an excessive

number of support vectors. Conversely, a very low value of C can lead to underfitting, where the model fails to capture the underlying data patterns adequately.

We chose the Radial Basis Function (RBF) kernel for our SVM model due to its ability to handle non-linear relationships in the data effectively. The parameter γ in the RBF kernel is equally important as it determines the influence range of support vectors. Specifically, γ can be seen as the inverse of the radius of influence for support vectors. A high γ value can cause the model to overfit by limiting the influence of support vectors to very narrow regions. On the other hand, a low γ value can lead to underfitting by restricting the model's capacity to assimilate information from the input data.

To identify the optimal combination of C and γ, we employed a "grid search" strategy. This method involves systematically exploring all possible pairs of C and γ values within a specified numeric range. Specifically, we examined C values ranging from 2^{-5} to 2^{15} and γ values ranging from 2^{3} to 2^{-15}, including intermediate values such as $2^{-4.5}$ and $2^{2.5}$. This comprehensive search allowed us to identify the most effective parameter pair for our model.

$\mathbf{GGL}_{XGBoost}$: In our experiments with the XGBoost algorithm, we aimed to optimize the model's performance by tuning its hyperparameters through a random search strategy. This approach allowed us to explore a wide range of parameter values efficiently. The resulting optimal parameters were as follows: number of estimators of 2000, a subsample rate of 0.8, a maximum depth of 3 for the trees, and a learning rate of 0.01. Additionally, we set the `colsample_bytree` to 0.6 and the `scale_pos_weight` to 5.

$\mathbf{GGL}_{CatBoost}$: In our work with CatBoost, we utilized a random search strategy to fine-tune the hyperparameters, aiming to achieve optimal model performance. The parameters we settled on included a learning rate of 0.05 and an L2 leaf regularization value of 1. We configured the model to run for 2000 iterations, with a tree depth set at 10. Additionally, the border count was set to 128, and the `scale_pos_weight` was adjusted to 3 to address class imbalance. We also disabled verbose output.

In addition to our primary approach, we also experimented with incorporating Morgan fingerprints generated by RDKit to enhance the feature set for our model for the $BBBP_{d_2}$ dataset. Morgan fingerprints, which are a type of circular fingerprint, provide a powerful way to represent molecular structures based on their connectivity. This representation captures local environments around each atom, making it a robust descriptor for cheminformatics applications. To integrate Morgan fingerprints with our geometric graph learning (GGL) features, we first generated the Morgan fingerprints using RDKit. These fingerprints encode the molecular structure in a high-dimensional binary vector format, which allows for the capture of intricate molecular features and patterns. Concurrently, we scaled our GGL features using the MinMaxScaler from scikit-learn, transforming the feature values to a range [0,1]. This scaling technique transforms the features to a common scale without distorting differences in the ranges of values. By doing so, we ensure that the GGL features are

normalized, facilitating a more effective combination with the Morgan fingerprints. After scaling the GGL features, we concatenated them with the Morgan fingerprints to create a comprehensive feature set that leverages both the structural and geometric information of the molecules. This combined feature vector was then used as the input for our machine learning models.

4.3 Data Collections and Preparations

In our research, we employed two separate datasets concerning Blood-Brain Barrier Permeability (BBBP). To facilitate clarity within our manuscript, we denote them as $BBBP_{d_1}$ and $BBBP_{d_2}$. The $BBBP_{d_1}$ dataset is a publicly available dataset that comprises a total of 2039 compounds, with 1560 categorized as positive (BBB+) and 479 as negative (BBB-). The dataset is collected from Xia *et al.* [111] and provided in SMILES string format. We then generate 3D coordinates of the compounds in MOL2 format using OpenEye toolkit [95].

Conversely, the $BBBP_{d_2}$ dataset contains a total of 1593 compounds, with 1283 classified as positive (BBB+) and 310 as negative (BBB-). Originally, the dataset was reported by Adenot *et al.* [2] and refined by Zhao *et al.* [144] in SMILES string format. We converted the compounds in 3D coordinates into MOL2 format using the OpenEye toolkit [95]. A summary of both datasets utilized in this study has been listed in Table4.1.

Table 4.1: Summary of datasets $BBBP_{d_1}$ and $BBBP_{d_2}$

Dataset	Total Compounds	# of BBB+	# of BBB-
$BBBP_{d_1}$	2039	1560	479
$BBBP_{d_2}$	1593	1283	310

To ensure the robust evaluation of our models, we employed a scaffold splitting method to partition the datasets into training, validation, and test sets. Scaffold splitting is a technique that groups structurally similar molecules together, ensuring that the training set includes diverse scaffolds. This method helps in evaluating the model's ability to generalize to novel chemical structures. We followed an 8:1:1 ratio for the splitting, as recommended in recent studies [111, 19]. This means that 80% of the data was used for training, 10% for validation, and 10% for testing.

The scaffold splitting approach provides a rigorous test of model performance by ensuring that the validation and test sets contain molecules with scaffolds not seen during training. This setup mimics real-world scenarios where the model encounters novel compounds, thus providing a realistic measure of its predictive power and generalizability. This method of splitting is particularly important in drug discovery and cheminformatics, where the ability to predict the properties of new, unseen molecules is crucial.

4.3.1 Evaluation Metrics

In assessing the predictive performance of models trained on the BBBP datasets ($BBBP_{d_1}$ and $BBBP_{d_2}$), we employ a set of evaluation metrics tailored to the specific characteristics of each dataset. Below is a comprehensive explanation of these metrics.

Accuracy measures the proportion of correctly classified instances among all instances. It considers both true positives (TP) and true negatives (TN) and is calculated by dividing the sum of TP and TN by the total number of instances.

$$\text{Accuracy (ACC)} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.13}$$

Sensitivity, also known as True Positive Rate or Recall, measures the proportion of actual positive cases that are correctly identified by the classifier. It is calculated by dividing TP by the sum of TP and false negatives (FN).

$$\text{Sensitivity (SN)} = \frac{TP}{TP + FN} \tag{4.14}$$

Specificity, also known as True Negative Rate, measures the proportion of actual negative cases that are correctly identified by the classifier. It is calculated by dividing TN by the sum of TN and false positives (FP).

$$\text{Specificity (SP)} = \frac{TN}{TN + FP} \tag{4.15}$$

The Matthews Correlation Coefficient (MCC) is a measure of the quality of binary classifications, considering all four values of the confusion matrix. It ranges from -1 to 1, where 1 indicates perfect prediction, 0 indicates random prediction, and -1 indicates total disagreement between prediction and observation. It takes into account TP, TN, FP, and FN to provide a balanced assessment of the classifier's performance.

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \tag{4.16}$$

AUC, or the Area Under the ROC Curve, is a common metric for evaluating classification performance that summarizes the model's ability to distinguish between positive and negative instances. To calculate AUC, we need to compute the True Positive Rate (TPR) and False Positive Rate (FPR). TPR measures the proportion of actual positives correctly classified, while FPR measures the proportion of actual negatives incorrectly classified as positives. The ROC curve plots TPR against FPR, and AUC summarizes the curve's performance, with higher values indicating better classification ability.

To maintain consistency with other state-of-the-art models, the performances of $BBBP_{d_1}$ dataset are primarily assessed using accuracy (ACC) and the Area Under the ROC curve (ROC-AUC). Conversely, for $BBBP_{d_2}$, model performance is reported based on accuracy (ACC), Matthews Correlation Coefficient (MCC).

4.4 Results and Discussion

In this section, we present the experimental results of GGL model on the datasets $BBBP_{d_1}$ and $BBBP_{d_2}$ coupled with various machine learning techniques to evaluate its performance in predicting blood-brain barrier permeability (BBBP).

4.4.1 Model Parametrization and Hyperparameter Optimization

In this section, we present the results of hyperparameter optimization and the performance evaluation of our proposed GGL-Score models. These models utilize both SYBYL atom-type weighted colored subgraph descriptors and atom-level descriptors on various BBBP datasets. Additionally, we conduct a rigorous similarity test analysis to validate the robustness of GGL-Score.

In our study, we use the notation ${}^{T}\mathrm{GGL}^{\mathcal{M}}_{\beta,\kappa,\tau}$ to describe geometric graph learning features. Here, $\mathcal{M}$ denotes the machine learning model used, which could be rf (Random Forest), CNN (Convolutional Neural Network), svm (Support Vector Machine), $XGBoost$, $CatBoost$ or Chemprop assessing DMPNN (Direct message passing neural network. For the Chemprop, we adapt the notations $Chem-atom$ and $Chem-mol$ for atom-level and molecule-level GGL features, respectively. The parameter β specifies the kernel type, while κ and τ are the respective kernel parameters. Specifically, we use the generalized exponential kernel, $\beta = E$, and the generalized Lorentz kernel, $\beta = L$ to generate GGL features for the cutoff distance $c = 12$Å. The parameter τ acts as a scaling factor that determines the characteristic distance $\eta_{kk'}$ between atom type k and atom type k'. This distance is calculated as $\eta_{kk'} = \tau(r_k + r_{k'})$, where r_k and $r_{k'}$ are the van der Waals radii of the atoms of type k and type k', respectively. Here, T indicates the type of the geometric graph learning features we generated. For instance, $T = ATOM$ signifies the geometric graph learning molecular atom features, whereas $T = SYBYL$ is the geometric graph learning extended atom-type weighted colored subgraph features.

Through hyperparameter optimization and model performance evaluation, we have shown that our GGL-Score models deliver robust and accurate predictions across various BBBP datasets. For each dataset used in this study, we optimize two critical parameters, κ and τ, for a given kernel type β to generate geometric graph learning features for extended atom types. We employ a five-fold cross-validation (CV) combined with a grid search method to identify the best values for $\tau \in (0.5, 10)$ and $\kappa \in (0.5, 10)$, incrementing both parameters by 0.5 during the search. Higher values for the power parameter κ are selected to approximate the ideal low-pass filter (ILF) [94].

For the GGL extended atom-type descriptors, we conducted a five-fold cross-validation on the training set, excluding the test set, to determine the optimal kernel parameters for each BBBP dataset. In the case of the $BBBP_{d_1}$ dataset, the best kernel parameters were identified as $\beta = L$, $\kappa = 20$, and $\tau = 0.5$, resulting in a median accuracy (out of 50 runs) of 0.91876 and an AUC of 0.94121. For the $BBBP_{d_2}$ dataset, the optimal kernel parameters were found to be $\beta = E$, $\kappa = 5.5$,

and $\tau = 0.5$, achieving a median accuracy (out of 50 runs) of 0.977, an AUC of 0.967, and an MCC of 0.861.

However, for GGL molecular atom features, we utilized a random choice of kernel parameters for both datasets rather than conducting a five-fold cross-validation. We utilize the GGL atom features only for the ${}^{ATOM}\text{GGL}^{Chem-atom}$ model.

4.4.2 $BBBP_{d_1}$ dataset

The experimental results of the GGL model on the $BBBP_{d_1}$ dataset reveal distinct performances across different model configurations. The ${}^{ATOM}\text{GGL}^{Chem-atom}_{E,2.5,1.5}$ model achieved an AUC of 0.938 and an accuracy (ACC) of 0.907, indicating high predictive performance by leveraging atom-level features. This suggests that detailed atomic interactions and spatial configurations captured by ${}^{ATOM}\text{GGL-Chem}^{atom}$ significantly enhance prediction accuracy.

In contrast, the ${}^{SYBYL}\text{GGL}^{Chem-mol}_{L,20,0.5}$ model, which focuses on molecule-level features, achieved an AUC of 0.890 and an ACC of 0.849. While still robust, the lower performance compared to the atom-level model implies that the finer granularity of atomic interactions offers a more accurate depiction of BBB permeability.

The ${}^{SYBYL}\text{GGL}^{rf}_{L,20,0.5}$ model, which integrates GGL features with a random forest classifier for the prediction task, demonstrated strong performance with an AUC of 0.921 and an ACC of 0.898. This result underscores the effectiveness of ensemble methods in handling complex, high-dimensional feature spaces derived from GGL.

On the other hand, the ${}^{SYBYL}\text{GGL}^{CNN}_{L,20,0.5}$ model, which utilizes GGL features with a convolutional neural network for prediction, obtained an AUC of 0.760 and an ACC of 0.771.

A summary of the performances on the $BBBP_{d_1}$ dataset is presented in Table 4.4.2.

Table 4.2: Performance on $BBBP_{d_1}$ dataset

Models	ACC	AUC
${}^{ATOM}\text{GGL}^{Chem-atom}_{E,2.5,1.5}$	0.907	0.938
${}^{SYBYL}\text{GGL}^{rf}_{L,20,0.5}$	0.898	0.921
${}^{SYBYL}\text{GGL}^{Chem-mol}_{L,20,0.5}$	0.849	0.890
${}^{SYBYL}\text{GGL}^{CNN}_{L,20,0.5}$	0.771	0.760

The performance comparison of various state-of-the-art models on the $BBBP_{d_1}$ dataset is presented in Figure 4.3. When compared with other state-of-the-art models, the GGL based models, particularly ${}^{ATOM}\text{GGL}^{Chem-atom}_{E,2.5,1.5}$ and ${}^{SYBYL}\text{GGL}^{rf}_{L,20,0.5}$, significantly outperformed the other models. The DMPNN model[136], while effective, did not match the performance of the GGL models, indicating that the detailed atom-type and geometric graph representations in GGL models capture more relevant features for predicting blood-brain barrier permeability. Similarly, comparing with other geometry-based models like Mol-GDL[111], and GEM[41] showed lower

performance, underscoring the advantage of GGL's detailed atom-type interactions over traditional methods.

The findings also demonstrate that graph neural networks and attention-based models like GROVE[105], PretrainGNN[55] and AttentiveFP[135], despite their sophistication, do not achieve the same level of performance as our GGL extended atom-type models. Which indicate that our GGL-Score fusion models as well as the the base GGL-score models significantly outperformed other reported performances on $BBBP_{d_1}$ dataset.

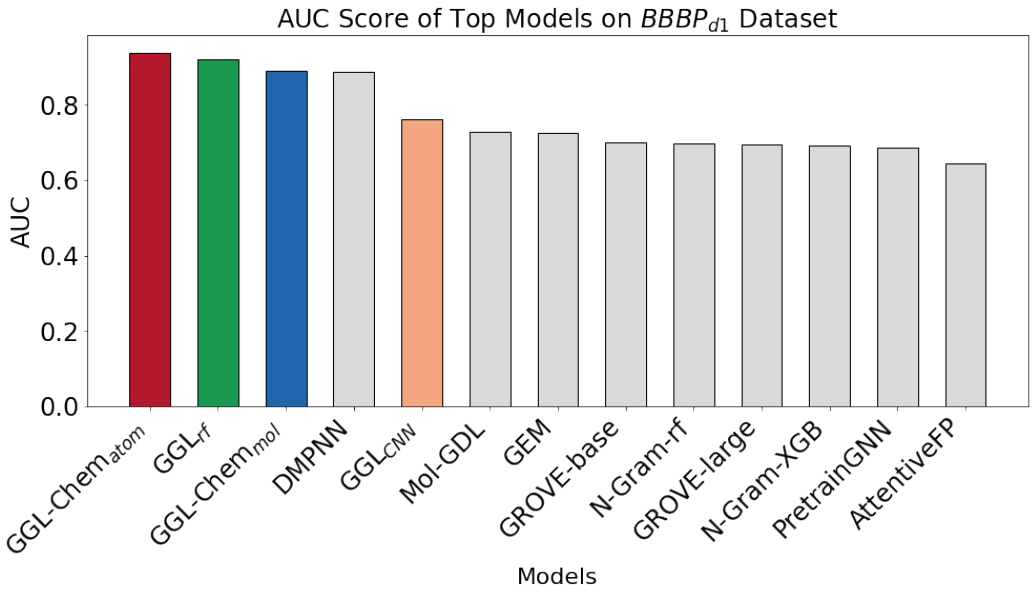

Figure 4.3: Performance comparison of different models on $BBBP_{d_1}$ dataset. The AUC scores of the other methods are collected from[136, 111]. The red, green, blue, and orange bars showcase the performances of our geometric graph learning-based models.

4.4.3 $BBBP_{d_2}$ dataset

The experimental results on $BBBP_{d_2}$, utilizing geometric extended atom-type graph features combined with various machine learning and deep learning models, are summarized in Table 4.3. We observe that the top performance is achieved by the ${}^{SYBYL}\text{GGL-Mfps}^{svm}_{E,5.5,0.5}$ model, which attains an MCC of 0.919. In contrast, ${}^{SYBYL}\text{GGL}^{svm}_{E,5.5,0.5}$ achieves an MCC of 0.876, indicating that the inclusion of Morgan fingerprints with the geometric graph features significantly enhances the predictive performance. Further comparison reveals that the ${}^{SYBYL}\text{GGL}^{Chem-atom}_{E,2.5,1.5}$ model achieved an MCC of 0.894, which is slightly lower than the ${}^{SYBYL}\text{GGL-Mfps}^{svm}_{E,5.5,0.5}$ model. This highlights the effectiveness of atom-level features in capturing essential chemical interactions. On the other hand, the ${}^{SYBYL}\text{GGL}^{Chem-mol}_{E,5.5,0.5}$ model, which focuses on molecule-level features, achieved an MCC of 0.870. The ${}^{SYBYL}\text{GGL}^{rf}_{E,5.5,0.5}$

model, with an MCC of 0.870, and its variant ^{SYBYL}GGL-Mfps$^{rf}_{E,5.5,0.5}$, with an MCC of 0.871, show that random forest models also benefit from the inclusion of Morgan fingerprints, although the improvement is marginal. Similarly, the ^{SYBYL}GGL$^{CatBoost}_{E,5.5,0.5}$ and ^{SYBYL}GGL-Mfps$^{CatBoost}_{E,5.5,0.5}$ models demonstrate MCCs of 0.872 and 0.884, respectively, further underscoring the positive impact of combining Morgan fingerprints with geometric graph features. Interestingly, the ^{SYBYL}GGL$^{XGBoost}_{E,5.5,0.5}$ model achieves an MCC of 0.882, while the ^{SYBYL}GGL-Mfps$^{XGBoost}_{E,5.5,0.5}$ model shows a slightly lower MCC of 0.854. This suggests that while the inclusion of Morgan fingerprints generally enhances performance, the extent of improvement can vary depending on the specific machine learning algorithm used.

Overall, these results highlight the superior performance of the ^{SYBYL}GGL-Mfps$^{svm}_{E,2.5,1.5}$ model, emphasizing the importance of incorporating detailed 2D molecular fingerprints with geometric graph features for predicting blood-brain barrier permeability. The enhancements provided by Morgan fingerprints across different machine learning models validate their significance in improving the accuracy and robustness of predictive models in cheminformatics.

Table 4.3: Summary of results $BBBP_{d_2}$ Dataset

Model	ACC	AUC	MCC
^{ATOM}GGL$^{Chem-atom}_{E,2.5,1.5}$	0.982	0.976	0.894
^{SYBYL}GGL$^{Chem-mol}_{E,5.5,0.5}$	0.978	0.953	0.870
^{SYBYL}GGL$^{svm}_{E,5.5,0.5}$	0.978	0.935	0.876
^{SYBYL}GGL-Mfps$^{svm}_{E,5.5,0.5}$	0.986	0.956	0.919
^{SYBYL}GGL$^{rf}_{E,5.5,0.5}$	0.978	0.972	0.870
^{SYBYL}GGL-Mfps$^{rf}_{E,5.5,0.5}$	0.978	0.970	0.871
^{SYBYL}GGL$^{CatBoost}_{E,5.5,0.5}$	0.979	0.960	0.872
^{SYBYL}GGL-Mfps$^{CatBoost}_{E,5.5,0.5}$	0.980	0.951	0.884
^{SYBYL}GGL$^{XGBoost}_{E,5.5,0.5}$	0.980	0.954	0.882
^{SYBYL}GGL-Mfp$^{XGBoost}_{E,5.5,0.5}$	0.974	0.959	0.854
Zhao et al [144]	0.972	-	0.844
Shen et al [112]	0.982	-	0.895

4.5 Conclusion

In this project, we implemented a ligand-only geometric graph learning (GGL) framework aimed at predicting blood-brain barrier permeability (BBBP). Our methodology began with the utilization of SYBYL extended atom types for ligand molecules to

create weighted colored subgraph descriptors. These descriptors were then integrated with a variety of machine learning and deep learning models to predict BBBP. In the second phase, we generated GGL molecular atom graph features and integrated these features with Chemprop's directed message-passing neural networks. This dual approach, leveraging both atom-type and molecular graph features, allowed us to achieve high accuracy in predicting blood-brain permeability using various evaluation metrics across the two BBBP datasets employed in this project.

The integration of GGL features with machine learning models and Chemprop's neural networks demonstrated significant predictive power and robustness. Our approach not only captured the geometric and chemical properties of ligands but also ensured accurate and reliable BBBP predictions across different datasets. The success of this project underscores the potential of geometric graph learning models in cheminformatics and their application in drug discovery and development.

Chapter 5 Conclusion and Discussion

In this dissertation, we developed and analyzed mathematical graph-based machine learning and deep learning models for extended atom types, incorporating weighted colored subgraphs. This innovative approach aims to enhance the predictive accuracy and robustness of molecular property predictions by leveraging the structural and chemical properties encoded within the graph representations of molecules.

In Chapter 1, we delve into the drug discovery pipeline in detail, highlighting the stages involved and the significance of each step. We also discuss the application of machine learning and deep learning models in drug discovery, emphasizing their transformative impact on the field.

Chapter 2 provides an overview of graphs and mathematical graph-based machine learning models, reviewing related work in this area. We then present our algebraic graph-based machine learning model for extended atom types (AGL-EAT-Score) and introduce the geometric graph-based ligand-only machine learning model for extended atom types in sections 2.3 and 2.4. These models are designed to capture the intricate details of molecular structures and their interactions, offering a more nuanced approach to property prediction.

In Chapter 3, we apply the AGL-EAT-Score model to predict protein-ligand binding affinity using several benchmark datasets, including CASF-2016, CASF-2013, and the CatS dataset. The performance of the model is rigorously evaluated, demonstrating its capability to provide accurate and reliable predictions across diverse datasets. We further validate the robustness and address concerns of overfitting in machine learning-based scoring functions, we tested the performance of the proposed AGL-EAT-Score against redundant and non-redundant data built on the PDBbind general set v2016 and the CatS dataset

Finally, in Chapter 4, we focus on predicting blood-brain barrier permeability. For this, we utilize a geometric graph learning model that incorporates extended atom type weighted colored subgraph descriptors, as well as geometric graph learning atom descriptors. We employ various machine learning models to evaluate the effectiveness of these descriptors in predicting the ability of compounds to cross the blood-brain barrier.

Overall, this dissertation provides a comprehensive exploration of graph-based machine learning and deep learning models for molecular property prediction. Through the development and application of innovative models, we demonstrate the potential of leveraging molecular properties encoded in molecular graphs to enhance predictive accuracy and robustness in various aspects of drug discovery.

Future Directions

- The methodologies developed in this dissertation can be extended to predict a broader range of molecular properties beyond binding affinity and blood-brain barrier permeability. Properties such as toxicity, solubility, and metabolic

stability are critical for drug discovery and can benefit from the application of graph-based learning models.

- Creating user-friendly software tools and platforms that implement these graph-based learning models could facilitate their adoption by researchers and practitioners. Such tools could provide accessible interfaces for data input, model training, and prediction output, enabling wider usage in the scientific community.

- Applying generative AI techniques with extended atom type color graphs within Graph Convolutional Networks (GCNs) to design new drugs. This approach uses GCNs to model molecular structures, where each atom type and its properties are represented in a colored graph format. By integrating generative models like Variational Autoencoders (VAEs) or Generative Adversarial Networks (GANs) with these GCNs, it becomes possible to generate novel molecular structures with desired properties. This integration enables the encoding of molecular graphs into latent space representations and decoding them to produce new molecules.

- Improving pose ranking accuracy in molecular docking studies. Pose ranking predicts the most likely binding conformation of a ligand within a target protein's active site, which is essential for understanding ligand-receptor interactions. Strategies to enhance pose ranking include developing better scoring functions that accurately capture physicochemical interactions and using advanced machine learning techniques to learn complex interaction patterns. Combining computational predictions with experimental data and using molecular dynamics (MD) simulations to explore conformational flexibility can also improve accuracy. Additionally, using graph-based representations of protein-ligand complexes and employing Graph Neural Networks (GNNs) for pose prediction and ranking can further enhance the reliability of docking studies.

Bibliography

[1] N. J. Abbott. Prediction of blood–brain barrier permeation in drug discovery from in vivo, in vitro and in silico models. *Drug Discovery Today: Technologies*, 1(4):407–416, 2004.

[2] M. Adenot and R. Lahana. Blood-brain barrier permeation models: discriminating between potential cns and non-cns drugs including p-glycoprotein substrates. *Journal of chemical information and computer sciences*, 44(1):239–248, 2004.

[3] D. D. Allen and Q. R. Smith. Characterization of the blood–brain barrier choline transporter using the in situ rat brain perfusion technique. *Journal of neurochemistry*, 76(4):1032–1041, 2001.

[4] N. J. Alves, S. D. Stimple, M. W. Handlogten, J. D. Ashley, T. Kiziltepe, and B. Bilgicer. Small-molecule-based affinity chromatography method for antibody purification via nucleotide binding site targeting. *Analytical chemistry*, 84(18):7721–7728, 2012.

[5] A. Angeleska, N. Jonoska, and M. Saito. Dna recombination through assembly graphs. *Discrete Applied Mathematics*, 157(14):3020–3037, 2009.

[6] R. E. Babine and S. L. Bender. Molecular recognition of protein- ligand complexes: Applications to drug design. *Chemical reviews*, 97(5):1359–1472, 1997.

[7] S. Bagchi, T. Chhibber, B. Lahooti, A. Verma, V. Borse, and R. D. Jayant. In-vitro blood-brain barrier models for drug screening and permeation studies: an overview. *Drug design, development and therapy*, pages 3591–3605, 2019.

[8] I. Bahar, A. R. Atilgan, and B. Erman. Direct evaluation of thermal fluctuations in proteins using a single-parameter harmonic potential. *Folding and Design*, 2(3):173–181, 1997.

[9] P. J. Ballester and J. B. Mitchell. A machine learning approach to predicting protein–ligand binding affinity with applications to molecular docking. *Bioinformatics*, 26(9):1169–1175, 2010.

[10] S. C. Basak, G. J. Niemi, and G. D. Veith. A graph-theoretic approach to predicting molecular properties. *Mathematical and Computer Modelling*, 14:511–516, 1990.

[11] B. A. Becker. Ligand–protein binding and screening using nmr spectroscopy. In J. C. Lindon, editor, *Encyclopedia of Spectroscopy and Spectrometry (Second Edition)*, pages 1322–1329. Academic Press, Oxford, second edition edition, 2010.

[12] C. M. Bishop. Pattern recognition and machine learning. *Springer google schola*, 2:1122–1128, 2006.

[13] K. H. Bleicher, H.-J. Böhm, K. Müller, and A. I. Alanine. Hit and lead generation: beyond high-throughput screening. *Nature reviews Drug discovery*, 2(5):369–378, 2003.

[14] D. Bramer and G.-W. Wei. Multiscale weighted colored graphs for protein flexibility and rigidity analysis. *The Journal of chemical physics*, 148(5):054103, 2018.

[15] A. Breda, N. F. Valadares, O. N. de Souza, and R. C. Garratt. Protein structure, modelling and applications. In *Bioinformatics in tropical disease research: a practical and case-study approach [Internet]*. National Center for Biotechnology Information (US), 2007.

[16] L. Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[17] Z. Cang and G.-W. Wei. Topologynet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions. *PLoS computational biology*, 13(7):e1005690, 2017.

[18] T. S. Carpenter, D. A. Kirshner, E. Y. Lau, S. E. Wong, J. P. Nilmeier, and F. C. Lightstone. A method to predict blood-brain barrier permeability of drug-like compounds using molecular dynamics simulations. *Biophysical journal*, 107(3):630–641, 2014.

[19] H. S. Chan, H. Shan, T. Dahoun, H. Vogel, and S. Yuan. Advancing drug discovery via artificial intelligence. *Trends in pharmacological sciences*, 40(8):592–604, 2019.

[20] O. Chapelle, B. Scholkopf, and A. Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

[21] D. Chen, K. Gao, D. D. Nguyen, X. Chen, Y. Jiang, G.-W. Wei, and F. Pan. Algebraic graph-assisted bidirectional transformers for molecular property prediction. *Nature Communications*, 12(1):1–9, 2021.

[22] T. Cheng, X. Li, Y. Li, Z. Liu, and R. Wang. Comparative assessment of scoring functions on a diverse test set. *Journal of chemical information and modeling*, 49(4):1079–1093, 2009.

[23] P. R. Costa, M. L. Acencio, and N. Lemke. A machine learning approach for genome-wide prediction of morbid and druggable human genes based on systems-level data. In *BMC genomics*, volume 11, pages 1–15. Springer, 2010.

[24] L. Cucullo, M. Hossain, E. Rapp, T. Manders, N. Marchi, and D. Janigro. Development of a humanized in vitro blood–brain barrier model to screen for brain penetration of antiepileptic drugs. *Epilepsia*, 48(3):505–516, 2007.

[25] L. Cucullo, N. Marchi, M. Hossain, and D. Janigro. A dynamic in vitro bbb model for the study of immune cell trafficking into the central nervous system. *Journal of Cerebral Blood Flow & Metabolism*, 31(2):767–777, 2011.

[26] C. Dagenais, A. Avdeef, O. Tsinman, A. Dudley, and R. Beliveau. P-glycoprotein deficient mouse in situ blood–brain barrier permeability and its prediction using an in combo pampa model. *European Journal of Pharmaceutical Sciences*, 38(2):121–137, 2009.

[27] R. Daneman and A. Prat. The blood–brain barrier. *Cold Spring Harbor perspectives in biology*, 7(1):a020412, 2015.

[28] B. Davis, K. Mcloughlin, J. Allen, and S. R. Ellingson. Quantifying overfitting potential in drug binding datasets. In *Computational Science–ICCS 2020: 20th International Conference, Amsterdam, The Netherlands, June 3–5, 2020, Proceedings, Part III 20*, pages 585–598. Springer, 2020.

[29] R. S. DeWitte and E. I. Shakhnovich. Smog: de novo design method based on simple, fast, and accurate free energy estimates. 1. methodology and supporting evidence. *Journal of the American Chemical Society*, 118(47):11733–11744, 1996.

[30] L. Di, E. H. Kerns, I. F. Bezar, S. L. Petusky, and Y. Huang. Comparison of blood–brain barrier permeability assays: in situ brain perfusion, mdr1-mdckii and pampa-bbb. *Journal of pharmaceutical sciences*, 98(6):1980–1991, 2009.

[31] J. A. DiMasi, H. G. Grabowski, and R. W. Hansen. Innovation in the pharmaceutical industry: new estimates of r&d costs. *Journal of health economics*, 47:20–33, 2016.

[32] X. Dong. Current strategies for brain drug delivery. *Theranostics*, 8(6):1481, 2018.

[33] J. A. Doudna and E. Charpentier. The new frontier of genome engineering with crispr-cas9. *Science*, 346(6213):1258096, 2014.

[34] J. Drews. Drug discovery: a historical perspective. *science*, 287(5460):1960–1964, 2000.

[35] S. Ekins, J. Mestres, and B. Testa. In silico pharmacology for drug discovery: methods for virtual ligand screening and profiling. *British journal of pharmacology*, 152(1):9–20, 2007.

[36] M. D. Eldridge, C. W. Murray, T. R. Auton, G. V. Paolini, and R. P. Mee. Empirical scoring functions: I. the development of a fast empirical scoring function to estimate the binding affinity of ligands in receptor complexes. *Journal of computer-aided molecular design*, 11:425–445, 1997.

[37] S. R. Ellingson, B. Davis, and J. Allen. Machine learning and ligand binding predictions: a review of data, methods, and obstacles. *Biochimica et Biophysica Acta (BBA)-General Subjects*, 1864(6):129545, 2020.

[38] L. Es. Initial sequencing and analysis of the human genome. *nature*, 409:860–921, 2001.

[39] L. Euler. Leonhard euler and the königsberg bridges. *Scientific American*, 189(1):66–72, 1953.

[40] T. J. Ewing, S. Makino, A. G. Skillman, and I. D. Kuntz. Dock 4.0: search strategies for automated molecular docking of flexible molecule databases. *Journal of computer-aided molecular design*, 15:411–428, 2001.

[41] X. Fang, L. Liu, J. Lei, D. He, S. Zhang, J. Zhou, F. Wang, H. Wu, and H. Wang. Geometry-enhanced molecular representation learning for property prediction. *Nature Machine Intelligence*, 4(2):127–134, 2022.

[42] A. Fire, S. Xu, M. K. Montgomery, S. A. Kostas, S. E. Driver, and C. C. Mello. Potent and specific genetic interference by double-stranded rna in caenorhabditis elegans. *nature*, 391(6669):806–811, 1998.

[43] U. Food and D. Administration. New drug application (nda), 2019. Accessed: 2024-05-24.

[44] J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[45] Z. Gaieb, S. Liu, S. Gathiaka, M. Chiu, H. Yang, C. Shao, V. A. Feher, W. P. Walters, B. Kuhn, and M. G. Rudolph. D3r grand challenge 2: blind prediction of protein–ligand poses, affinity rankings, and relative binding free energies. *Journal of computer-aided molecular design*, 32(1):1–20, 2018.

[46] H. Gao, Z. Pang, and X. Jiang. Targeted delivery of nano-therapeutics for major disorders of the central nervous system. *Pharmaceutical research*, 30:2485–2498, 2013.

[47] E. Gawehn, J. A. Hiss, and G. Schneider. Deep learning in drug discovery. *Molecular Informatics*, 35(1):3–14, 2016.

[48] R. Gramatica, T. Di Matteo, S. Giorgetti, M. Barbiani, D. Bevec, and T. Aste. Graph theory enables drug repurposing–how a mathematical model can drive the discovery of hidden mechanisms of action. *PloS one*, 9(1):e84912, 2014.

[49] P. J. Hansen and P. C. Jurs. Chemical applications of graph theory. part i. fundamentals and topological indices. *J. Chem. Educ*, 65(7):574, 1988.

[50] M. Hartenfeller, H. Zettl, M. Walter, M. Rupp, F. Reisen, E. Proschak, S. Weggen, H. Stark, and G. Schneider. Dogs: reaction-driven de novo design of bioactive compounds. *PLoS computational biology*, 8(2):e1002380, 2012.

[51] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[52] P. C. Hawkins, A. G. Skillman, and A. Nicholls. Comparison of shape-matching and docking as virtual screening tools. *Journal of medicinal chemistry*, 50(1):74–82, 2007.

[53] E. Heid, K. P. Greenman, Y. Chung, S.-C. Li, D. E. Graff, F. H. Vermeire, H. Wu, W. H. Green, and C. J. McGill. Chemprop: A machine learning package for chemical property prediction. *Journal of Chemical Information and Modeling*, 64(1):9–17, 2023.

[54] T. Hou, J. Wang, and Y. Li. Adme evaluation in drug discovery. 8. the prediction of human intestinal absorption by a support vector machine. *Journal of chemical information and modeling*, 47(6):2408–2415, 2007.

[55] W. Hu, B. Liu, J. Gomes, M. Zitnik, P. Liang, V. Pande, and J. Leskovec. Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265*, 2019.

[56] C.-H. Huang, P. M.-H. Chang, C.-W. Hsu, C.-Y. F. Huang, and K.-L. Ng. Drug repositioning for non-small cell lung cancer by using machine learning algorithms and topological graph theory. In *BMC bioinformatics*, volume 17, pages 13–26. BioMed Central, 2016.

[57] F. Imrie, T. E. Hadfield, A. R. Bradley, and C. M. Deane. Deep generative design with 3d pharmacophoric constraints. *Chemical science*, 12(43):14577–14589, 2021.

[58] D. Janezic, A. Milicevic, S. Nikolic, and N. Trinajstic. *Graph-theoretical matrices in chemistry*. CRC Press, 2015.

[59] I. Jarmoskaite, I. AlSadhan, P. P. Vaidyanathan, and D. Herschlag. How to measure and evaluate binding affinities. *Elife*, 9:e57264, 2020.

[60] J. Jeon, S. Nim, J. Teyra, A. Datti, J. L. Wrana, S. S. Sidhu, J. Moffat, and P. M. Kim. A systematic approach to identify novel cancer drug targets using machine learning, inhibitor design and high-throughput screening. *Genome medicine*, 6:1–18, 2014.

[61] J. Jiang, R. Wang, M. Wang, K. Gao, D. D. Nguyen, and G.-W. Wei. Boosting tree-assisted multitask deep learning for small scientific datasets. *Journal of chemical information and modeling*, 60(3):1235–1244, 2020.

[62] G. Jones, P. Willett, R. C. Glen, A. R. Leach, and R. Taylor. Development and validation of a genetic algorithm for flexible docking. *Journal of molecular biology*, 267(3):727–748, 1997.

[63] H. Jones, Y. Chen, C. Gibson, T. Heimbach, N. Parrott, S. Peters, J. Snoeys, V. Upreti, M. Zheng, and S. Hall. Physiologically based pharmacokinetic modeling in drug discovery and development: a pharmaceutical industry perspective. *Clinical Pharmacology & Therapeutics*, 97(3):247–262, 2015.

[64] H. Kadry, B. Noorani, and L. Cucullo. A blood–brain barrier overview on structure, function, impairment, and biomarkers of integrity. *Fluids and Barriers of the CNS*, 17:1–24, 2020.

[65] S. L. Kinnings, N. Liu, P. J. Tonge, R. M. Jackson, L. Xie, and P. E. Bourne. A machine learning-based method to improve docking scoring functions and its application to drug repurposing. *Journal of chemical information and modeling*, 51(2):408–419, 2011.

[66] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[67] S. Kocbek and J.-D. Kim. Exploring biomedical ontology mappings with graph theory methods. *PeerJ*, 5:e2990, 2017.

[68] G. Landrum. Rdkit: Open-source cheminformatics, 2006.

[69] K. Lanevskij, J. Dapkunas, L. Juska, P. Japertas, and R. Didziapetris. Qsar analysis of blood–brain distribution: the influence of plasma and brain tissue binding. *Journal of pharmaceutical sciences*, 100(6):2147–2160, 2011.

[70] A. Lavecchia and C. Di Giovanni. Virtual screening strategies in drug discovery: a critical review. *Current medicinal chemistry*, 20(23):2839–2860, 2013.

[71] A. Lesk. Introduction to protein architecture (book) oxford university press, 2001.

[72] H. Li, K.-S. Leung, M.-H. Wong, and P. J. Ballester. Improving autodock vina using random forest: the growing accuracy of binding affinity prediction by the effective exploitation of larger data sets. *Molecular informatics*, 34(2-3):115–126, 2015.

[73] H. Li, J. Peng, Y. Leung, K.-S. Leung, M.-H. Wong, G. Lu, and P. J. Ballester. The impact of protein structure and sequence similarity on the accuracy of machine-learning scoring functions for binding affinity prediction. *Biomolecules*, 8(1):12, 2018.

[74] H. Li, J. Peng, P. Sidorov, Y. Leung, K.-S. Leung, M.-H. Wong, G. Lu, and P. J. Ballester. Classical scoring functions for docking are unable to exploit large volumes of structural and interaction data. *Bioinformatics*, 35(20):3989–3995, 2019.

[75] H. Li, C. W. Yap, C. Y. Ung, Y. Xue, Z. W. Cao, and Y. Z. Chen. Effect of selection of molecular descriptors on the prediction of blood- brain barrier penetrating and nonpenetrating agents by statistical learning methods. *Journal of Chemical Information and Modeling*, 45(5):1376–1384, 2005.

[76] Y. Li, L. Han, Z. Liu, and R. Wang. Comparative assessment of scoring functions on an updated benchmark: 2. evaluation methods and general results. *Journal of chemical information and modeling*, 54(6):1717–1736, 2014.

[77] Y. Li and J. Yang. Structural and sequence similarity makes a significant impact on machine-learning-based scoring functions for protein–ligand interactions. *Journal of chemical information and modeling*, 57(4):1007–1012, 2017.

[78] C. A. Lipinski, F. Lombardo, B. W. Dominy, and P. J. Feeney. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Advanced drug delivery reviews*, 23(1-3):3–25, 1997.

[79] Z. Liu, Y. Li, L. Han, J. Li, J. Liu, Z. Zhao, W. Nie, Y. Liu, and R. Wang. Pdb-wide collection of binding data: current status of the pdbbind database. *Bioinformatics*, 31(3):405–412, 2015.

[80] Z. Liu, M. Su, L. Han, J. Liu, Q. Yang, Y. Li, and R. Wang. Forging the basis for developing protein–ligand interaction scoring functions. *Accounts of Chemical Research*, 50(2):302–309, 2017.

[81] J. M. Luco. Prediction of the brain- blood distribution of a large set of drugs from structurally derived descriptors using partial least-squares (pls) modeling. *Journal of chemical information and computer sciences*, 39(2):396–404, 1999.

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

[83] G. Maggiora, M. Vogt, D. Stumpfe, and J. Bajorath. Molecular similarity in medicinal chemistry: miniperspective. *Journal of medicinal chemistry*, 57(8):3186–3204, 2014.

[84] Z. Meng and K. Xia. Persistent spectral–based machine learning (perspect ml) for protein-ligand binding affinity prediction. *Science advances*, 7(19):eabc5329, 2021.

[85] J. Mensch, J. Oyarzabal, C. Mackie, and P. Augustijns. In vivo, in vitro and in silico methods for small molecule transfer across the bbb. *Journal of pharmaceutical sciences*, 98(12):4429–4468, 2009.

[86] R. Miao, L.-Y. Xia, H.-H. Chen, H.-H. Huang, and Y. Liang. Improved classification of blood-brain-barrier drugs using deep learning. *Scientific reports*, 9(1):8802, 2019.

[87] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[88] G. M. Morris, R. Huey, W. Lindstrom, M. F. Sanner, R. K. Belew, D. S. Goodsell, and A. J. Olson. Autodock4 and autodocktools4: Automated docking with selective receptor flexibility. *Journal of computational chemistry*, 30(16):2785–2791, 2009.

[89] I. Muegge and Y. C. Martin. A general and fast scoring function for protein-ligand interactions: a simplified potential approach. *Journal of medicinal chemistry*, 42(5):791–804, 1999.

[90] D. Nagarajan and N. Chandra. Pocketmatch (version 2.0): A parallel algorithm for the detection of structural similarities between protein ligand binding-sites. In *2013 National Conference on Parallel Computing Technologies (PARCOMPTECH)*, pages 1–6. IEEE, 2013.

[91] S. Nakagawa, M. A. Deli, H. Kawaguchi, T. Shimizudani, T. Shimono, A. Kittel, K. Tanaka, and M. Niwa. A new blood–brain barrier model using primary rat brain endothelial cells, pericytes and astrocytes. *Neurochemistry international*, 54(3-4):253–263, 2009.

[92] S. Nakagawa, M. A. Deli, S. Nakao, M. Honda, K. Hayashi, R. Nakaoke, Y. Kataoka, and M. Niwa. Pericytes from brain microvessels strengthen the barrier integrity in primary cultures of rat brain endothelial cells. *Cellular and molecular neurobiology*, 27:687–694, 2007.

[93] D. D. Nguyen and G.-W. Wei. Agl-score: algebraic graph learning score for protein–ligand binding scoring, ranking, docking, and screening. *Journal of chemical information and modeling*, 59(7):3291–3304, 2019.

[94] D. D. Nguyen, T. Xiao, M. Wang, and G.-W. Wei. Rigidity strengthening: A mechanism for protein–ligand binding. *Journal of chemical information and modeling*, 57(7):1715–1721, 2017.

[95] OpenEye Scientific Software. http://www.eyesopen.com, 2023. OpenEye Scientific Software, Santa Fe, NM.

[96] OpenEye Scientific Software. ROCS 3.5.1.1. http://www.eyesopen.com, 2023. OpenEye Scientific Software, Santa Fe, NM.

[97] W. M. Pardridge. Blood–brain barrier delivery. *Drug discovery today*, 12(1-2):54–61, 2007.

[98] C. D. Parks, Z. Gaieb, M. Chiu, H. Yang, C. Shao, W. P. Walters, J. M. Jansen, G. McGaughey, R. A. Lewis, and S. D. Bembenek. D3r grand challenge 4: blind prediction of protein–ligand poses, affinity rankings, and relative binding free energies. *Journal of computer-aided molecular design*, 34(2):99–119, 2020.

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

[100] G. A. Pavlopoulos, M. Secrier, C. N. Moschopoulos, T. G. Soldatos, S. Kossida, J. Aerts, R. Schneider, and P. G. Bagos. Using graph theory to analyze biological networks. *BioData mining*, 4:1–27, 2011.

[101] W. R. Pearson and D. J. Lipman. Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences*, 85(8):2444–2448, 1988.

[102] M. M. Rana and D. D. Nguyen. Geometric graph learning with extended atom-types features for protein-ligand binding affinity prediction. *arXiv preprint arXiv:2301.06194*, 2023.

[103] M. Randić and C. L. Wilkins. Graph-theoretical analysis of molecular properties. isomeric variations in nonanes. *International Journal of Quantum Chemistry*, 18(4):1005–1027, 1980.

[104] H. P. Rang, M. M. Dale, J. M. Ritter, R. J. Flower, and G. Henderson. *Rang & Dale's pharmacology*. Elsevier Health Sciences, 2011.

[105] Y. Rong, Y. Bian, T. Xu, W. Xie, Y. Wei, W. Huang, and J. Huang. Self-supervised graph transformer on large-scale molecular data. *Advances in neural information processing systems*, 33:12559–12571, 2020.

[106] F. Sams-Dodd. Target-based drug discovery: is something wrong? *Drug discovery today*, 10(2):139–147, 2005.

[107] G. Schneider. Virtual screening: an endless staircase? *Nature Reviews Drug Discovery*, 9(4):273–276, 2010.

[108] H. P. Schultz. Topological organic chemistry. 1. graph theory and topological indices of alkanes. *Journal of Chemical Information and Computer Sciences*, 29(3):227–228, 1989.

[109] J. H. Seo, C. S. Kim, and H. J. Cha. Structural evaluation of gm1-related carbohydrate–cholera toxin interactions through surface plasmon resonance kinetic analysis. *Analyst*, 138(22):6924–6929, 2013.

[110] O. Shalem, N. E. Sanjana, and F. Zhang. High-throughput functional genomics using crispr–cas9. *Nature Reviews Genetics*, 16(5):299–311, 2015.

[111] C. Shen, J. Luo, and K. Xia. Molecular geometric deep learning. *Cell Reports Methods*, 3(11), 2023.

[112] J. Shen, F. Cheng, Y. Xu, W. Li, and Y. Tang. Estimation of adme properties with substructure pattern recognition. *Journal of chemical information and modeling*, 50(6):1034–1041, 2010.

[113] J. Shen, Y. Du, Y. Zhao, G. Liu, and Y. Tang. In silico prediction of blood–brain partitioning using a chemometric method called genetic algorithm based variable selection. *QSAR & Combinatorial Science*, 27(6):704–717, 2008.

[114] R. B. Silverman and M. W. Holladay. *The organic chemistry of drug design and drug action*. Academic press, 2014.

[115] C. J. Stam and J. C. Reijneveld. Graph theoretical analysis of complex networks in the brain. *Nonlinear biomedical physics*, 1:1–19, 2007.

[116] M. M. Stepniewska-Dziubinska, P. Zielenkiewicz, and P. Siedlecki. Development and evaluation of a deep learning model for protein-ligand binding affinity prediction. *Bioinformatics*, 1:9, 2018.

[117] M. Su, G. Feng, Z. Liu, Y. Li, and R. Wang. Tapping on the black box: how is the scoring power of a machine-learning scoring function dependent on the training set? *Journal of chemical information and modeling*, 60(3):1122–1136, 2020.

[118] M. Su, Q. Yang, Y. Du, G. Feng, Z. Liu, Y. Li, and R. Wang. Comparative assessment of scoring functions: the casf-2016 update. *Journal of chemical information and modeling*, 59(2):895–913, 2018.

[119] Q. Tang, F. Nie, Q. Zhao, and W. Chen. A merged molecular representation deep learning method for blood–brain barrier permeability prediction. *Briefings in Bioinformatics*, 23(5):bbac357, 2022.

[120] Z. Tong, J. E. Schiel, E. Papastavros, C. M. Ohnmacht, Q. R. Smith, and D. S. Hage. Kinetic studies of drug–protein interactions by using peak profiling and high-performance affinity chromatography: examination of multi-site interactions of drugs with human serum albumin columns. *Journal of Chromatography A*, 1218(15):2065–2071, 2011.

[121] N. Trinajstic. Chemical graph theory. *Boca Raton*, 1983.

[122] N. Trinajstic. *Chemical graph theory*. Routledge, 2018.

[123] O. Trott and A. J. Olson. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of Computational Chemistry*, 31(2):455–461, 2010.

[124] J. Vamathevan and et al. Applications of machine learning in drug discovery and development. *Nat Rev Drug Discov*, 18(6):463–477, 2019.

[125] V. Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995.

[126] H. F. Velec, H. Gohlke, and G. Klebe. Drugscorecsd knowledge-based scoring function derived from small molecule crystal data with superior recognition rate of near-native ligand poses and better affinity prediction. *Journal of medicinal chemistry*, 48(20):6296–6303, 2005.

[127] G. Verkhivker, K. Appelt, S. Freer, and J. Villafranca. Empirical free energy calculations of ligand-protein crystallographic complexes. i. knowledge-based ligand-protein interaction potentials applied to the prediction of human immunodeficiency virus 1 protease binding affinity. *Protein Engineering, Design and Selection*, 8(7):677–691, 1995.

[128] A. Vilella, B. Ruozi, D. Belletti, F. Pederzoli, M. Galliani, V. Semeghini, F. Forni, M. Zoli, M. A. Vandelli, and G. Tosi. Endocytosis of nanomedicines: the case of glycopeptide engineered plga nanoparticles. *Pharmaceutics*, 7(2):74–89, 2015.

[129] B. Wang, Z. Zhao, D. D. Nguyen, and G.-W. Wei. Feature functional theory–binding predictor (fft–bp) for the blind prediction of binding free energies. *Theoretical Chemistry Accounts*, 136(4):1–22, 2017.

[130] C. Wang and Y. Zhang. Improving scoring-docking-screening powers of protein–ligand scoring functions using random forest. *Journal of computational chemistry*, 38(3):169–177, 2017.

[131] R. Wang, D. D. Nguyen, and G.-W. Wei. Persistent spectral graph. *International journal for numerical methods in biomedical engineering*, 36(9):e3376, 2020.

[132] Z. Wang, H. Yang, Z. Wu, T. Wang, W. Li, Y. Tang, and G. Liu. In silico prediction of blood–brain barrier permeability of compounds by machine learning and resampling methods. *ChemMedChem*, 13(20):2189–2201, 2018.

[133] B. Wei, Y. Zhang, and X. Gong. Deeplpi: a novel deep learning-based model for protein–ligand interaction prediction for drug repurposing. *Scientific reports*, 12(1):18200, 2022.

[134] H. Wiener. Structural determination of paraffin boiling points. *Journal of the American Chemical Society*, 69(1):17–20, 1947.

[135] Z. Xiong, D. Wang, X. Liu, F. Zhong, X. Wan, X. Li, Z. Li, X. Luo, K. Chen, H. Jiang, et al. Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism. *Journal of medicinal chemistry*, 63(16):8749–8760, 2019.

[136] K. Yang, K. Swanson, W. Jin, C. Coley, P. Eiden, H. Gao, A. Guzman-Perez, T. Hopper, B. Kelley, M. Mathea, et al. Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, 59(8):3370–3388, 2019.

[137] L. W. Yang and C.-P. Chng. Coarse-grained models reveal functional dynamics—i. elastic network models—theories, comparisons and perspectives. *Bioinformatics and Biology Insights*, 2:25–45, 2008.

[138] K. Yeturu and N. Chandra. Pocketmatch: a new algorithm to compare binding sites in protein structures. *BMC bioinformatics*, 9:1–17, 2008.

[139] R. C. Young, R. C. Mitchell, T. H. Brown, C. R. Ganellin, R. Griffiths, M. Jones, K. K. Rana, D. Saunders, and I. R. Smith. Development of a new physicochemical model for brain penetration and its application to the design of centrally acting h2 receptor histamine antagonists. *Journal of medicinal chemistry*, 31(3):656–671, 1988.

[140] Y. Yuan, F. Zheng, and C.-G. Zhan. Improved prediction of blood–brain barrier permeability through machine learning with combined use of molecular property-based descriptors and fingerprints. *The AAPS journal*, 20:1–10, 2018.

[141] C. Yung-Chi and W. H. Prusoff. Relationship between the inhibition constant (ki) and the concentration of inhibitor which causes 50 per cent inhibition (i50) of an enzymatic reaction. *Biochemical pharmacology*, 22(23):3099–3108, 1973.

[142] J. Zah, G. Terre'Blanche, E. Erasmus, and S. F. Malan. Physicochemical prediction of a brain–blood distribution profile in polycyclic amines. *Bioorganic & medicinal chemistry*, 11(17):3569–3578, 2003.

[143] W. Zhang, G. Sigdel, K. J. Mintz, E. S. Seven, Y. Zhou, C. Wang, and R. M. Leblanc. Carbon dots: A future blood–brain barrier penetrating nanomedicine and drug nanocarrier. *International journal of nanomedicine*, pages 5003–5016, 2021.

[144] Y. H. Zhao, M. H. Abraham, A. Ibrahim, P. V. Fish, S. Cole, M. L. Lewis, M. J. de Groot, and D. P. Reynolds. Predicting penetration across the blood-brain barrier from simple descriptors and fragmentation schemes. *Journal of chemical information and modeling*, 47(1):170–175, 2007.

[145] Z. Zheng and K. M. Merz Jr. Ligand identification scoring algorithm (lisa). *Journal of chemical information and modeling*, 51(6):1296–1306, 2011.