

Geometric Deep Learning in Drug Discovery

by

Yan Sun

A thesis submitted to the Faculty of Graduate Studies of
The University of Manitoba
in partial fulfillment of the requirements of the degree of

Doctor of Philosophy

Department of Computer Science
University of Manitoba
Winnipeg

Copyright © 2025 by Yan Sun

Abstract

Artificial intelligence (AI) is transforming early-stage drug discovery by enabling efficient, data-driven molecular modeling and prediction. In this thesis, we present a series of interpretable and task-specific methods grounded in Geometric Deep Learning (GDL) to advance molecular representation learning across key domains such as molecular property prediction and drug–target interaction (DTI).

We introduced four original methods that address core challenges in molecular representation learning. These methods target modality alignment, multiscale feature integration, interpretable analysis, and target imbalance in regression. Together, they enable the learning of robust and geometry-aware molecular representations, supporting diverse downstream tasks under limited or imbalanced data conditions.

To illustrate practical utility, we apply our models to a preliminary virtual screening task for PIN1 (peptidyl-prolyl cis-trans isomerase NIMA-interacting 1), an oncogenic driver in cancer. Using a curated dataset with potency and efficacy annotations, we define a composite activity metric and show the ability of the model to prioritize active compounds.

Together, these contributions demonstrate the versatility of GDL-based approaches in addressing various molecular learning tasks and their potential to enable interpretable, high-performance AI frameworks for accelerating drug discovery.

Keywords: drug discovery, geometric deep learning, molecular representation, target imbalance, molecular property prediction, drug-target interaction prediction

Acknowledgements

I would like to express my deepest gratitude to my parents for their unwavering support, both financial and emotional, throughout my academic journey.

To all of my friends, thank you for being part of my life. Your presence has been my greatest fortune. I am especially grateful for the encouragement, companionship, and sense of security you offered during the most difficult times. I sincerely wish each of you a life full of love, fulfillment, and self-realization.

I would like to thank my supervisor, Dr. Pingzhao Hu, for the guidance, trust, and freedom that allowed me to grow as a researcher. I am also grateful to Dr. Carson K Leung for his academic support during my graduate studies. To my research group colleagues, thank you for the valuable discussions and the joy of sharing ideas together. Those moments have been truly rewarding.

I am deeply grateful to my therapist, whose support has brought me immense comfort and contributed significantly to my personal growth.

I am thankful for every experience along the way.

Finally, I express all my longing for my grandmother. I never imagined I would never see you again, but now, more than a year has passed, and your phone will never ring again.

Contents

Abstract	ii
Acknowledgements	iii
List of Figures	ix
List of Tables	xii
List of Abbreviations	1
1 Introduction	2
1.1 Drug Discovery and Its Challenges	2
1.2 Computational Methods in Drug Discovery	4
1.3 The Role of Deep Learning in Drug Discovery	8
1.4 Geometric Deep Learning: A Natural Fit	9
1.4.1 Geometric Priors	9
1.4.2 Graph Neural Networks	12
1.5 Geometric Deep Learning for Molecular Representation	13
1.5.1 Supervised Models for Molecular Representation	14
1.5.2 Chemical Language Models for Molecular Representation	16
1.6 Aims of this Thesis	18
1.7 Thesis Format and Publications	19

2	Molecular Property Prediction based on Bimodal Supervised Contrastive Learning	21
2.1	Introduction	22
2.2	Related Work	24
2.3	The Proposed BSCL Method	24
2.3.1	Problem Formulation	25
2.3.2	Model Architecture	25
2.3.3	Loss Function	25
2.3.4	Including Global Feature	29
2.4	Experiments	31
2.4.1	Dataset and Evaluation	31
2.4.2	Baselines	31
2.4.3	Parameter Setting	32
2.4.4	Result and analysis	32
2.5	Summary	35
3	MolGraph-xLSTM: A graph-based dual-level xLSTM framework with multi-head mixture-of-experts for enhanced molecular representation and interpretability	37
3.1	Introduction	38
3.2	Background and Related Work	41
3.2.1	Extended Long Short-Term Memory	41
3.2.2	Deep Learning Methods Based on Molecular Graphs for representation learning to predict molecular properties	43
3.2.3	Multi-Head Mixture of Experts	43
3.3	Method	45
3.3.1	Construction of Atom- and Motif-Level Molecular Graphs	45
3.3.2	Feature Extraction on Atom-Level Graph	46

3.3.3	Motif-Level Feature Extraction	49
3.3.4	Perform a Multi-Head Mixture-of-Experts on the Features	49
3.3.5	Overall architecture	49
3.3.6	Loss function	50
3.4	Experiments	53
3.4.1	Datasets and evaluation	53
3.4.2	Baselines	55
3.4.3	Experimental results	58
3.4.4	Ablation Study	63
3.4.5	Interpretability analysis	68
3.4.6	Hyperparameter analysis	70
3.5	Summary	71
4	DistRouting: Distribution-Aware Expert Routing Guided by Physicochemical Descriptors for Imbalanced Molecular Property Regression	73
4.1	Introduction	74
4.2	Related Work	76
4.3	Method: DistRouting	78
4.3.1	Preliminaries and Notation	78
4.3.2	Distribution-aware Routing Guided by Physicochemical Descriptors	78
4.3.3	Gating Supervision with Soft Target Labels	79
4.3.4	Interval-Aware Supervised Contrastive Learning	80
4.3.5	Prediction and Overall Loss	81
4.4	Experiments	82
4.4.1	Experimental Setting	82
4.4.2	Overall Performance	85
4.4.3	Region-wise Performance	87
4.4.4	Ablation Study	90

4.5	Discussion	91
4.6	Summary	94
5	iNGNN-DTI: Prediction of Drug - Target Interaction with Interpretable Nested Graph Neural Network and Pretrained Molecule Models	95
5.1	Introduction	96
5.2	Materials and Methods	98
5.2.1	Datasets	98
5.2.2	Input Data Representation	99
5.3	Methods	101
5.3.1	k-subgraph GNN extractor layer with virtual node	102
5.3.2	Interaction module	104
5.3.3	Merge features extracted from pre-trained models	105
5.4	Baseline Models and Model Evaluation	106
5.5	Results	106
5.5.1	Performance for DTI prediction	106
5.5.2	Performance on the Test Set with Unseen Drugs and Targets	107
5.5.3	Interpretability Analysis	108
5.5.4	Ablation Study	111
5.5.5	COVID-19 Case Study	115
5.6	Summary	117
6	Case Study: Discovery of PIN1 Inhibitors	120
6.1	Introduction	120
6.2	Materials and Methods	123
6.2.1	Datasets	123
6.2.2	PIN1 inhibitor screening	124
6.3	Experiment	124

6.3.1	Experiment Setting	124
6.3.2	Screening Results	125
6.4	Discussion	126
6.5	Summary	128
7	Conclusion and Future Work	129
7.1	Conclusion	129
7.2	Future Work	130
	References	131

List of Figures

1.1	Overview of the drug discovery and development pipeline.	3
1.2	Computational tasks in early-stage drug discovery.	5
1.3	Multimodal representations of small molecules and proteins.	10
1.4	Illustration of invariance and equivariance.	11
1.5	Message Passing Neural Network (MPNN) framework.	12
2.1	Model architecture and joint embedding space	23
2.2	t-SNE visualization on the BACE dataset	27
2.3	t-SNE visualization on the ESOL dataset	33
2.4	RMSE vs. γ_1 and γ_2 on Freesolv	34
3.1	Comparison of atom-level and motif-level graph representations.	39
3.2	Architecture of MolGraph-xLSTM.	45
3.3	Ablation study results on the Sider and FreeSolv datasets	62
3.4	Activation patterns of samples across the experts in the MHMoE module on the Freesolv dataset.	64
3.5	Activation patterns of samples across the experts in the MHMoE module on the Sider dataset.	65
3.6	Examples of different atom input orders for molecular graphs in xLSTM.	66
3.7	Performance comparison of MolGraph-xLSTM trained with different node or- derings.	66
3.8	Visualization of the highest-weighted motifs and atoms identified by the model for molecules from the Sider test set containing the SO_2NH substructure.	67

3.9	Importance scores of substructures identified by MolGraph-xLSTM on the BBBP dataset.	68
3.10	Analysis of the impact of the number of experts and heads on the Sider and FreeSolv datasets.	69
3.11	Performance of the model with varying numbers of jump layers.	71
4.1	DistRouting framework and routing mechanism.	76
4.2	ISCL mechanism.	80
4.3	Target distributions of the datasets.	83
4.4	MAE on the validation set over training epochs.	85
4.5	Region-wise MAE across four datasets and three backbone models.	88
4.6	t-SNE visualization of molecular representations.	90
4.7	Distribution matching across datasets.	92
4.8	Expert assignment distribution on LD50 dataset.	93
5.1	The process of the generation of protein graph with AlphaFold2.	100
5.2	Model architecture of iNGNN-DTI.	102
5.3	Visualization of the residues with the highest weight in each dimension on the target structure.	109
5.4	Grid box configuration for virtual docking on 2YFX.	110
5.5	Visualization of the virtual docking result.	111
5.6	Visualization of the atoms with the highest weight in each dimension on the drug molecule.	112
5.7	ROC and PR curves comparing two contact map generation methods on the DAVIS dataset: AlphaFold2 vs. SPOT-Contact-LM.	113
5.8	Comparison of pLDDT scores between membrane and non-membrane proteins across three datasets. Proteins are classified as high-confidence (pLDDT > 70) or low-confidence (pLDDT ≤ 70).	113

5.9	Visualization of the alignment between AlphaFold-predicted structures (blue) and corresponding PDB experimental structures (green).	114
5.10	Virtual docking scores for top 50 drugs, and top 50-100 drugs.	116
5.11	Visualization of grid box-based virtual docking results for drugs DB00786 and DB00616 on the ACE2 target.	117
5.12	Comparison of blind docking results and our grid box-based virtual docking results.	118
6.1	Mechanism of Pin1-mediated cis–trans isomerization and its inhibition, adapted from [1].	121
6.2	Workflow of PIN1 inhibitor screening.	124
6.3	Predicted binding poses of top candidate molecules within the PIN1 active site.	126
6.4	2D interaction diagrams of top candidate molecules with PIN1.	127

List of Tables

2.1	Dataset details.	29
2.2	Hyperparameters for each dataset.	30
2.3	Performance for regression tasks and classification tasks.	30
2.4	Ablation test on the BACE and Freesolv datasets.	35
3.1	Atom features and descriptions.	46
3.2	Bond features and descriptions.	46
3.3	Node features of motif-level graph.	47
3.4	Details of the datasets from MoleculeNet.	53
3.5	Details of datasets from TDC.	53
3.6	Hyperparameter setting for datasets from MoleculeNet.	54
3.7	Hyperparameter setting for datasets from TDC.	54
3.8	Performance evaluation on classification datasets from MoleculeNet using AU-ROC and AUPRC.	56
3.9	Performance evaluation on classification datasets from MoleculeNet using Accuracy and F1 score.	56
3.10	Performance evaluation on regression datasets from MoleculeNet using RMSE and Pearson Correlation (PCC).	57
3.11	Performance evaluation on regression datasets from MoleculeNet using MAE and coefficient of determination (R2).	57
3.12	Performance evaluation on classification datasets from TDC using AUROC. . .	59
3.13	Performance evaluation on classification datasets from TDC using AUPRC. . .	59

3.14	Performance evaluation on classification datasets from TDC using Accuracy. . .	60
3.15	Performance evaluation on classification datasets from TDC using F1 score. . .	60
3.16	Performance evaluation on regression datasets from TDC using RMSE and Pearson Correlation (PCC).	61
3.17	Performance evaluation on regression datasets from TDC using MAE and R2. .	61
3.18	Ablation results for MolGraph-xLSTM on the Sider (classification) and Free- Solv (regression) datasets.	63
4.1	Overview of the datasets.	83
4.2	Hyperparameter settings.	85
4.3	MAE ($\downarrow$) on the four datasets.	85
4.4	PCC ($\uparrow$) on the four datasets.	86
4.5	Region-wise MAE ($\downarrow$) across all datasets and methods.	87
4.6	Region-wise PCC ($\uparrow$) across all datasets and backbones.	89
4.7	Ablation study of DistRouting components.	90
4.8	Validation MAE under different ISCL weights λ	91
4.9	JS distance ($\downarrow$) between predicted and true target distributions.	91
5.1	Summary of the datasets	99
5.2	Atom features in the drug molecule.	99
5.3	Protein molecular graph feature initialization.	101
5.4	Model performance comparison.	107
5.5	Model performance on the test set with unseen drugs and proteins.	108
5.6	Ablation test on the DAVIS dataset.	111
5.7	Performance on the DAVIS dataset with varied cut-offs for binarization of the contact map.	115
6.1	Final candidate compounds for PIN1 inhibition after virtual screening.	125

List of Abbreviations

ADMET	Absorption, Distribution, Metabolism, Excretion, and Toxicity
AFT	Attention-Free Transformer
BERT	Bidirectional Encoder Representations from Transformers
BRICS	Breaking of Retrosynthetically Interesting Chemical Substructures
DTI	Drug–Target Interaction
GDL	Geometric Deep Learning
GNN	Graph Neural Network
HTS	High-Throughput Screening
JK	Jumping Knowledge (for GNNs)
MoE	Mixture-of-Experts
PE	Potency-to-Efficacy Ratio
PIN1	Peptidyl-prolyl cis-trans isomerase NIMA-interacting 1
PPM	Position Probability Matrix
PSSM	Position-Specific Scoring Matrix
SAR	Structure–Activity Relationship
SCL	Supervised Contrastive Learning
SMILES	Simplified Molecular Input Line Entry System
VS	Virtual Screening
xLSTM	Extended Long Short-Term Memory

Chapter 1

Introduction

1.1 Drug Discovery and Its Challenges

Drugs are biologically active compounds that exert their effects by binding to specific molecular structures in the human body, commonly referred to as targets. These targets are typically proteins such as enzymes, receptors, ion channels, or transporters [2]. Through this interaction, drugs can modulate the biological activity of the target, either by inhibiting or activating it, thereby altering physiological pathways and producing therapeutic effects. This molecular mechanism forms the foundation of modern pharmacology and provides the basis for rational drug design.

The development of a new drug is a complex, time-consuming, and resource-intensive process. It typically begins with the identification and validation of a disease-relevant, drug-gable target, followed by the discovery and optimization of lead compounds, which are small molecules with promising biological activity and potential for development into effective therapeutics. These leads undergo preclinical evaluation and are later tested in clinical trials before regulatory approval. The entire pipeline is often divided into two major phases: *drug discovery*, which includes target validation, lead identification, and lead optimization; and *drug development*, which comprises preclinical and clinical testing as well as regulatory review.

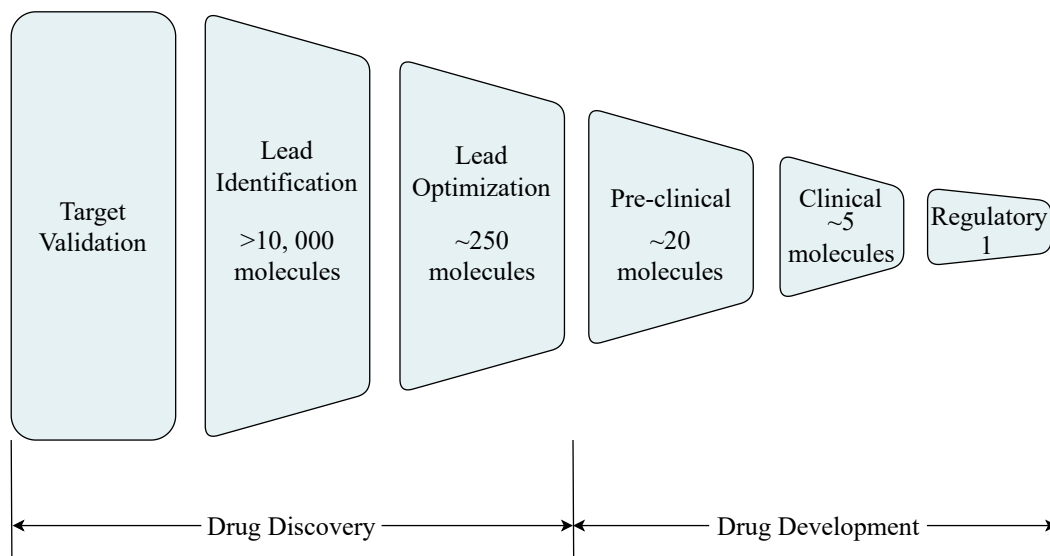


Figure 1.1: **Overview of the drug discovery and development pipeline.** The drug development process progresses from target validation to regulatory approval, with a substantial reduction in candidate compounds at each stage. The funnel is broadly divided into drug discovery and drug development phases.

This process is commonly illustrated as a narrowing funnel, reflecting the progressive elimination of candidate compounds through successive stages of development (Figure 1.1). According to recent industry estimates, it takes on average 10 to 15 years and over \$2 billion USD to bring a new drug to market, with overall success rates from beginning to regulatory approval remaining below 10% [3]. Such inefficiencies pose both scientific and economic burdens, especially as demands for targeted therapies, personalized medicine, and rapid response to emerging diseases continue to rise.

A variety of scientific and technical challenges contribute to the systematic elimination of candidate compounds throughout the drug discovery process. A central difficulty lies in the sheer size of chemical space: with an estimated 10^{60} synthetically accessible small molecules, exhaustive exploration through experimental screening is infeasible [4]. Even when promising leads are identified, optimizing their potency, selectivity, pharmacokinetics, and safety remains a highly iterative and resource-intensive task. Chemical modifications that improve on-target activity may simultaneously exacerbate off-target liabilities, such as hERG channel inhibition, which is a major contributor to drug-induced cardiotoxicity and clinical attrition.

Approximately 60% of drug candidates exhibit measurable hERG block, and nearly 30% of drug withdrawals between 1990 and 2006 were associated with QT prolongation and Torsades de Pointes [5]. This reflects the difficulty of balancing permeability, bioavailability, and safety, as molecules with favorable pharmacokinetic properties such as high lipophilicity and positive charge also tend to bind the hERG channel.

Clinical translation presents additional challenges. Many compounds that show promising results in early assays ultimately fail in clinical trials, especially during Phase II, where therapeutic efficacy is assessed. A study by Waring et al. identified insufficient efficacy and unexpected toxicity as the primary causes of failure [6], often due to incomplete target validation or undetected off-target effects. Despite ongoing efforts to improve early prediction, existing models still struggle to capture key molecular complexities such as polypharmacology, absorption, distribution, metabolism, excretion, and toxicity (ADMET) properties, and chemical structure diversity. These limitations underscore the need for more effective and generalizable predictive frameworks.

To support early-stage screening and reduce experimental workload, high-throughput screening (HTS) has been widely adopted. It enables automated testing of large compound libraries using miniaturized assays, often beginning with a single-dose primary screen followed by confirmatory testing. However, HTS remains costly and inefficient, with low hit rates, high false-positive rates, and considerable assay variability [7]. To address these limitations, computational methods have been developed to prioritize candidates before experimental testing. These approaches aim to improve screening efficiency, reduce cost, and focus validation efforts on more promising compounds.

1.2 Computational Methods in Drug Discovery

To support early-stage compound screening and reduce experimental costs, a wide range of computational methods have been developed to prioritize candidate molecules before labora-

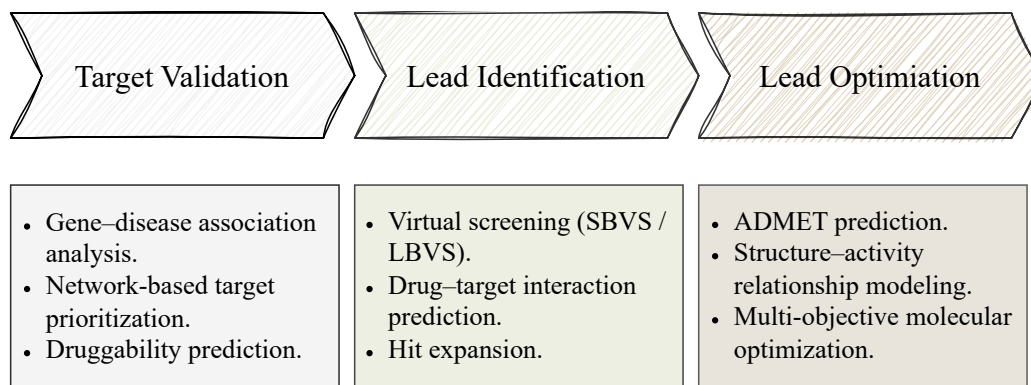


Figure 1.2: **Computational tasks in early-stage drug discovery.** Key computational methods supporting target validation, lead identification, and lead optimization.

tory testing. These methods are applied across the key phases of the drug discovery process, including target validation, lead identification, and lead optimization (Figure 1.2). They are used to predict molecular activity, physicochemical properties, and potential liabilities, thereby improving efficiency and reducing attrition.

During the target validation phase, computational tools are used to identify and prioritize disease-relevant, druggable targets. Typical tasks include analyzing gene–disease associations using omics data [8], modeling protein–protein interaction networks to detect key disease nodes [9, 10, 11], and predicting druggability based on sequence or structural features [12, 13]. For example, integrative multi-omics analyses have identified TREM2, a microglia-expressed receptor involved in innate immunity, as a key risk gene and potential therapeutic target in Alzheimer’s disease. Genome-wide association studies (GWAS), transcriptomics, and single-cell sequencing data have collectively highlighted TREM2’s role in modulating neuroinflammation and amyloid plaque clearance [14]. In parallel, tools such as DrugnomeAI [12] aggregate features from protein–protein interaction networks, essentiality scores, and structural data to prioritize novel druggable genes across the human genome. Machine learning models also support the identification of potential biomarkers and the integration of multi-modal biological data for target selection.

In the lead identification phase, computational methods assist in screening large chemical libraries to identify molecules likely to interact with the validated targets. A widely adopted

strategy is *virtual screening* (VS), which ranks candidate molecules based on their predicted binding potential. VS approaches are broadly categorized into structure-based and ligand-based methods [15].

Structure-based virtual screening (SBVS) relies on the three-dimensional (3D) structure of the target protein to assess ligand binding. It typically involves two steps: *pose generation*, which samples possible ligand conformations within the binding site, and *scoring*, which estimates binding affinity [15, 16, 17]. Popular docking tools such as AutoDock Vina [18], Glide [19], and GOLD [20] are commonly used in SBVS and implement heuristic or exhaustive search algorithms to explore ligand binding poses within the target binding site.

SBVS has contributed to several successful drug discovery campaigns [21, 22, 23]. For example, Chidambaram et al. [21] employed a combined E-pharmacophore and structure-based virtual screening pipeline to identify novel BACE-1 inhibitors for Alzheimer's disease. Two hits, ZINC39592220 and en1003sfl.46293, demonstrated favorable docking scores, stable binding to the catalytic dyad residues (ASP32 and ASP228), and good predicted pharmacokinetic properties. However, despite these successes, SBVS methods depend heavily on the availability of high-quality protein structures, struggle with receptor and ligand flexibility, and often rely on simplified scoring functions that may not fully capture solvation effects, entropic changes, or induced fit [24].

Ligand-based virtual screening (LBVS), in contrast, infers compound activity by comparing candidate molecules to known actives. It assumes that structurally similar compounds exhibit similar bioactivity. LBVS methods typically use molecular fingerprints, substructure patterns, or physicochemical descriptors to represent molecules, followed by similarity-based ranking or machine learning classifiers [25, 26]. Recent tools such as VSFlow [27] combine substructure, fingerprint, and 3D shape-based screening into a unified LBVS pipeline. These approaches are effective when sufficient ligand data are available, but tend to generalize poorly to novel targets or unfamiliar scaffolds.

Other lead identification tasks include drug–target interaction (DTI) prediction using com-

putational models that represent both molecular and protein features. Traditional DTI prediction methods can be broadly categorized into several classes, including feature-based classification, matrix factorization, neighborhood models, bipartite local models, and network diffusion approaches [28]. These methods leverage chemical similarity, interaction patterns, and graph topology to infer novel drug–target associations, and have formed the foundation for many modern machine learning applications in virtual screening. For example, SimBoost [29] employs gradient boosting on similarity-derived features to predict binding affinities, while DTINet [30] integrates heterogeneous biological networks through network diffusion and low-dimensional embeddings to uncover previously unobserved interactions.

In the lead optimization phase, computational methods are used to refine the chemical properties of candidate molecules and enhance their developability. This includes the prediction of ADMET properties, which are major contributors to late-stage failure [2]. Traditional rule-based heuristics such as Lipinski’s Rule of Five [31] and descriptor-based statistical models like FAF-Drugs2 [32] and VolSurf [33] have been used to estimate solubility, permeability, and blood–brain barrier penetration. More advanced tasks include structure–activity relationship (SAR) modeling to understand how molecular modifications impact bioactivity, and multi-objective optimization to balance competing goals such as potency, selectivity, and safety.

Although conventional computational methods have played an important role in early-stage drug discovery, they face notable limitations. Many rely on predefined descriptors and shallow models, which may fail to capture complex structure–activity relationships or generalize across diverse targets and chemical scaffolds. These challenges have motivated the development of data-driven approaches that can learn task-specific representations directly from raw molecular and biological data. In particular, deep learning has emerged as a powerful framework for modeling nonlinear, high-dimensional relationships and integrating heterogeneous inputs. The following section provides an overview of deep learning methods and their potential to address the limitations of traditional approaches.

1.3 The Role of Deep Learning in Drug Discovery

Over the past decade, deep learning has emerged as a transformative approach in many scientific and industrial domains, demonstrating state-of-the-art performance in fields such as computer vision, natural language processing, and speech recognition. The ability of deep neural networks to model complex, nonlinear relationships and learn task-relevant representations directly from raw data has motivated their adoption in computational biology and drug discovery [34].

In drug discovery, deep learning provides several notable advantages over traditional machine learning approaches. First, it reduces the dependence on manual feature engineering by automatically learning representations from raw molecular data. Second, it facilitates the integration of diverse data types, including chemical structures, bioassay results, protein information, and omics profiles, within a unified predictive framework. Third, deep learning models are highly scalable and can be trained on large datasets comprising millions of compounds.

These advantages have led to the rapid adoption of deep learning across a variety of tasks in drug discovery. For instance, convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have been successfully applied to molecular property prediction [35, 36]. Deep multitask networks developed by AstraZeneca have demonstrated improved accuracy in predicting ADMET properties compared to traditional quantitative structure–activity relationship (QSAR) methods [35]. Such successes highlight the practical value of deep learning in guiding early-stage compound selection and optimization.

Deep learning has also shown effectiveness in drug–target interaction (DTI) prediction by modeling both molecular and protein sequences to infer binding relationships [37, 38, 39, 40, 41]. For example, DeepDTA [37] uses convolutional neural networks to encode SMILES and protein sequences, while MolTrans [40] leverages transformer-based encoders to learn interaction patterns between molecular substructures and protein segments. These models have consistently outperformed traditional machine learning approaches such as SimBoost on benchmark datasets like KIBA [42] and Davis [43], demonstrating superior accuracy in predicting

binding affinities.

In addition, generative models such as variational autoencoders (VAEs), generative adversarial networks (GANs), and reinforcement learning frameworks have been explored for virtual screening and de novo molecular design [44, 45]. A particularly transformative application is in structural biology. AlphaFold2 [46] predicts protein 3D structures from amino acid sequences with near-experimental accuracy, significantly advancing structure-based drug discovery and target validation.

Despite the progress made by deep learning in drug discovery, several important challenges remain unresolved. Many current models operate on simplified representations such as 1D SMILES strings or molecular fingerprints, which may fail to capture the underlying structural and spatial complexity of molecules. As a result, their ability to generalize to novel scaffolds or preserve critical symmetries such as rotation, translation, and atom ordering is limited. Furthermore, deep models often lack inductive biases that reflect domain knowledge in chemistry or physics, making them less interpretable and data-efficient, especially in low-data settings.

To address these limitations, geometric deep learning has emerged as a promising framework for molecular and structural modeling. By representing molecules and proteins as graphs or geometric objects and incorporating symmetry priors directly into model design, GDL offers a more principled and expressive approach to learning from non-Euclidean chemical data. In the next section, we introduce the foundations of geometric deep learning and explain why it is well-suited for drug discovery applications.

1.4 Geometric Deep Learning: A Natural Fit

1.4.1 Geometric Priors

Molecular structures are inherently non-Euclidean. Unlike sequences, molecules are more naturally represented as graphs, where atoms are nodes and bonds are edges, or as three-dimensional geometries that reflect spatial and structural organization (Figure 1.3). These

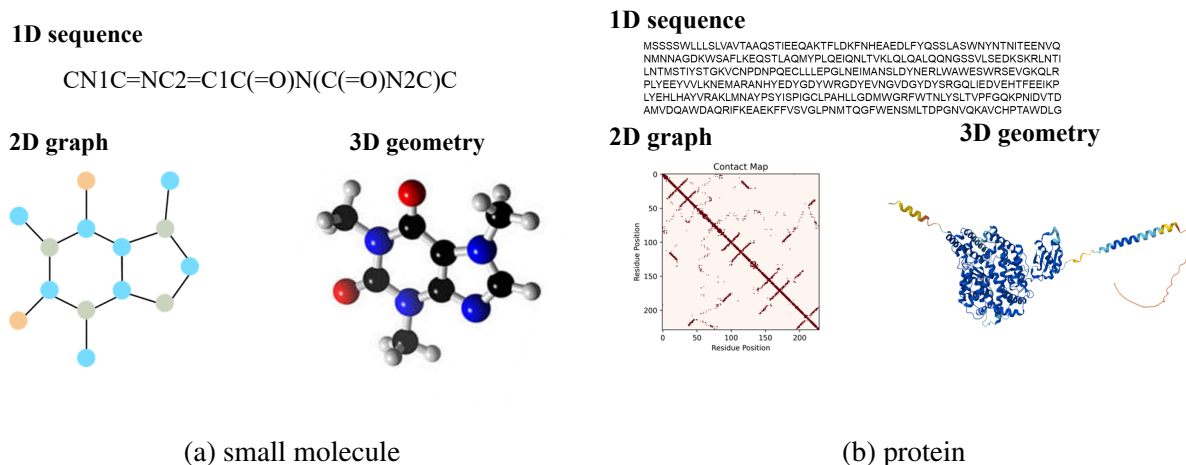


Figure 1.3: **Multimodal representations of (a) small molecules and (b) proteins.** Each biomolecule can be described at three levels: a 1D sequence, a 2D graph, and a 3D geometry.

representations encode complex relational and geometric information, making them incompatible with conventional deep learning architectures such as convolutional or recurrent neural networks, which operate on Euclidean grids or sequential data.

Geometric deep learning (GDL) provides a unified framework for extending neural networks to non-Euclidean domains such as graphs, manifolds, and meshes [47]. A central principle in GDL is the incorporation of *geometric priors*, which are domain-specific inductive biases such as symmetry and scale separation, directly into the model architecture.

Symmetries, including invariance and equivariance under permutations, rotations, and translations, are especially important in molecular modeling. For instance, predicting solubility should be invariant to atomic reordering or molecular orientation, whereas force prediction or pose estimation should yield outputs that vary consistently with the input configuration.

Formally, a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ is *invariant* to a transformation $\mathcal{T}$ if

$$f(\mathcal{T}(x)) = f(x), \quad (1.1)$$

for all $x \in \mathcal{X}$. In this case, the output remains unchanged under the transformation.

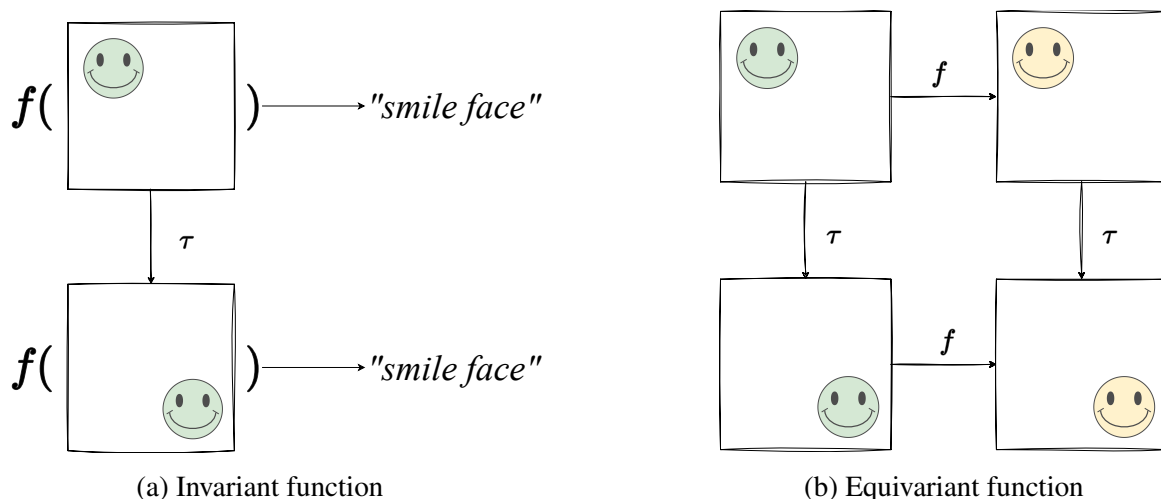


Figure 1.4: **Illustration of invariance and equivariance.** Invariant functions produce outputs that remain unchanged under transformations of the input, while equivariant functions produce outputs that transform consistently with the input. These symmetry properties are critical in geometric deep learning, particularly in molecular modeling tasks such as property prediction and force estimation.

In contrast, f is *equivariant* if

$$f(\mathcal{T}(x)) = \mathcal{T}(f(x)). \quad (1.2)$$

This means that applying $\mathcal{T}$ before or after f yields consistent results.

Figure 1.4 illustrates these concepts: invariant functions produce outputs that are unaffected by input transformations, while equivariant functions produce outputs that transform in the same way as the inputs. Such symmetry constraints enhance generalization and lead to more physically meaningful predictions in geometric and structural modeling tasks.

Another key geometric prior is *scale separation*, which encourages architectures to model systems hierarchically by first aggregating local information and then progressively integrating it into global representations. This is particularly relevant in molecular graphs, where local environments influence atom-level properties such as reactivity, while global structure affects higher-level behaviors such as binding affinity. A typical GDL pipeline includes local equivariant operators, global invariant pooling, and optionally coarsening mechanisms for hierarchical

abstraction.

1.4.2 Graph Neural Networks

GNNs operationalize geometric principles through the message-passing framework [48], in which each node iteratively aggregates information from its neighbors and updates its representation across layers.

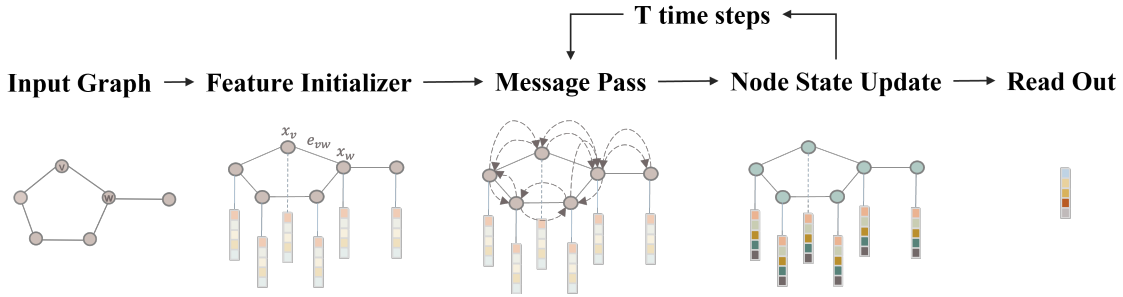


Figure 1.5: **Message Passing Neural Network (MPNN) framework.** The input graph is initialized with node and edge features. At each message passing step, nodes aggregate information from their neighbors and update their states. After multiple iterations, a readout function produces a global graph-level representation.

The message passing process typically consists of three stages: message computation, node update, and readout. At each iteration t , the hidden state of a node v is updated based on the aggregated messages from its neighbors $\mathcal{N}(v)$:

$$\begin{aligned} \mathbf{m}_v^t &= \text{AGG}_{t-1} \left(\{M_{t-1}(\mathbf{h}_v^{t-1}, \mathbf{h}_w^{t-1}, \mathbf{e}_{vw}) \mid w \in \mathcal{N}(v)\} \right), \\ \mathbf{h}_v^t &= U_{t-1}(\mathbf{h}_v^{t-1}, \mathbf{m}_v^t), \end{aligned} \tag{1.3}$$

where M is the message function, AGG is a permutation-invariant aggregation function (e.g., sum or mean), and U is the node update function.

After T iterations, node embeddings are aggregated into a graph-level representation via a readout function:

$$\mathbf{g} = R(\{\mathbf{h}_v^T \mid v \in G\}). \tag{1.4}$$

GNN variants primarily differ in the design of message, aggregation, and update functions. For example, Graph Convolutional Networks (GCNs) [49] use normalized mean aggregation, while Graph Attention Networks (GATs) [50] learn attention weights over neighbors:

$$M(\cdot) = \alpha_{vw} \mathbf{W} \mathbf{h}_w^{t-1}, \quad \alpha_{vw} = \frac{\exp(\text{LeakyReLU}(\mathbf{a}^\top [\mathbf{W} \mathbf{h}_v \| \mathbf{W} \mathbf{h}_w]))}{\sum_{u \in \mathcal{N}(v)} \exp(\text{LeakyReLU}(\mathbf{a}^\top [\mathbf{W} \mathbf{h}_v \| \mathbf{W} \mathbf{h}_u]))}, \quad (1.5)$$

where α_{vw} is the attention coefficient and $\|$ denotes vector concatenation.

Recent developments include gated update mechanisms (e.g., GRUs) [51], residual connections [52], and expressive aggregation schemes such as Principal Neighborhood Aggregation (PNA) [53]. These extensions enhance model depth, stability, and expressiveness, solidifying GNNs as a core architecture in geometric deep learning.

Beyond their algorithmic formulation, message-passing GNNs naturally encode key geometric priors. *Symmetry* is captured through permutation invariance, enforced by symmetric aggregation functions such as sum, mean, or max, ensuring node updates are independent of neighbor ordering. *Scale separation* is achieved by local message propagation, enabling the model to capture fine-grained structural patterns and progressively integrate them into higher-level representations across layers. These properties make GNNs particularly well-suited for learning over non-Euclidean domains such as molecular graphs and biomolecular structures.

1.5 Geometric Deep Learning for Molecular Representation

Geometric deep learning provides a unified framework for learning from structured data across multiple modalities and dimensions. In the context of drug discovery, it enables the integration of molecular information from one-dimensional (1D) sequences, two-dimensional (2D) graphs, and three-dimensional (3D) geometries. This flexibility is especially valuable for modeling both small molecules and proteins, which naturally exhibit structure at all three levels.

For small molecules, the 1D representation is typically the SMILES string, a linear textual encoding of the molecular graph. The 2D representation corresponds to the molecular graph

itself, with atoms as nodes and chemical bonds as edges. The 3D representation includes the spatial coordinates of atoms. Similarly, for proteins, 1D refers to the amino acid sequence, 2D can be represented as graphs using residue-level contact maps, where edges reflect spatial or functional proximity, and 3D refers to the full atomic or backbone coordinates in space.

In this section, we present geometric learning approaches across these representations, grouped into two modeling paradigms. The first focuses on supervised models trained directly on task-specific datasets. The second introduces chemical language models, which are typically pretrained on large-scale unlabeled data and fine-tuned for downstream applications.

1.5.1 Supervised Models for Molecular Representation

Supervised models for molecular representation are typically trained directly on labeled datasets to optimize for specific prediction tasks, such as molecular property estimation or drug-target interaction (DTI) prediction.

In the 1D setting, supervised models based on CNNs and RNNs have shown strong performance in extracting local and sequential features from these sequences. For instance, CNNs are widely adopted to capture substructural motifs in SMILES for property prediction tasks [54, 55], while RNN-based encoders such as LSTMs or GRUs model long-range dependencies in protein sequences for function classification or binding site identification [56, 57]. These architectures are computationally efficient and effective on moderate-sized datasets. Representative examples include DeepDTA, which employs parallel CNNs for SMILES and protein sequences to predict binding affinity [37], and DeepTTC, which uses LSTM encoders combined with dense feature fusion to model complex interactions [58]. More recent transformer-based architectures have been used to encode SMILES strings or fingerprint-like representations. Models such as FP-BERT [59] and Transformer-CNN [60] learn contextual embeddings of chemical substructures and have achieved strong performance on benchmark datasets.

In the 2D setting, graph neural networks (GNNs) have become a standard approach for learning structure-aware molecular representations. D-MPNN introduces directed bond-level

message passing and achieves strong results on benchmark datasets [61]. AttentiveFP leverages gated attention mechanisms to refine atom-level features, improving performance and interpretability across property prediction tasks [62]. Models such as GCN [49], GAT [50], and GraphSAGE [63] have been widely adapted to molecular graphs and serve as building blocks for multi-task and low-resource learning settings. In the context of drug–target interaction prediction, GraphDTA [64] and MGraphDTA [65] combine molecular graphs with protein sequences or contact maps to capture cross-modality interactions. Hybrid GNN architectures that incorporate node and edge features, hierarchical pooling, and global readout schemes further improve the model’s capacity to handle complex molecular structures and diverse predictive tasks.

In the 3D setting, molecules and proteins are modeled based on their spatial atomic coordinates. These representations enable models to capture geometric and physicochemical interactions crucial for molecular recognition. Early models such as SchNet [66] and DimeNet [67] incorporate interatomic distances and angles through continuous-filter convolutions or directional message passing. More recent approaches, such as EGNN [68] and SE(3)-Transformer [69], further improve expressiveness by ensuring equivariance to 3D transformations. On the protein side, models like GVP-GNN [70] leverage both scalar and directional vector features to encode geometric properties of protein surfaces, aiding in structure-based tasks such as interface prediction and binding pocket modeling. Although 3D supervised models are computationally demanding, they provide the most fine-grained structural detail and achieve superior performance in spatially sensitive tasks.

A notable real-world application of AI-driven molecular graph learning is INS018_055, a TNIK inhibitor developed by Insilico Medicine for treating idiopathic pulmonary fibrosis (IPF). Using a generative GNN-based design pipeline, the molecule was identified, optimized, and validated across in vitro and in vivo models, and subsequently advanced into Phase I clinical trials within just 18 months [71]. This work demonstrates the clinical viability of graph-based models for de novo drug discovery.

Together, these supervised models offer a versatile modeling toolkit that spans multiple structural resolutions. Depending on task requirements and data availability, one can choose from lightweight sequence models, expressive graph networks, or spatially aware 3D architectures to build tailored prediction pipelines. Their interpretability, efficiency, and direct optimization for downstream objectives make them a fundamental component in modern AI-driven drug discovery.

1.5.2 Chemical Language Models for Molecular Representation

Chemical language models leverage large-scale self-supervised pretraining to learn representations of molecules and proteins from sequence and structure data. Inspired by advances in natural language processing, these models treat molecular inputs such as SMILES strings or protein sequences as symbolic languages, and learn contextualized embeddings through masked token prediction, contrastive learning, or generative objectives. Unlike supervised models trained for a specific task, chemical language models are pretrained on massive unlabeled corpora and subsequently fine-tuned for diverse downstream applications in drug discovery.

In the 1D setting, protein sequences and SMILES strings are the primary modeling targets. Transformer-based architectures such as BERT [72] and T5 [73] have been adapted to these domains. On the protein side, models like ESM [74] and ProtBERT [75] are pretrained on millions to billions of amino acid sequences using masked language modeling or sequence-to-sequence learning. These models produce residue-level embeddings that capture evolutionary, structural, and functional features, enabling strong performance on tasks such as mutational effect prediction, ligand binding inference, and interface classification. On the molecular side, models such as MolBERT [76], ChemBERTa [77], and SMILES-BERT [78] are trained on large chemical libraries, generating atom-level embeddings that encode chemical syntax and semantics. These models are widely applied to molecular property prediction, toxicity classification, and scaffold-based generalization, often outperforming descriptor- or graph-based models in low-data regimes.

Although chemical language models are most naturally suited for 1D inputs, several extensions have been proposed to incorporate 2D structural information. GROVER [79] combines context prediction and motif prediction objectives through self-supervised learning on molecular graphs, enabling the extraction of transferable chemical knowledge across tasks. MolCLR [80] employs contrastive learning with chemically meaningful graph augmentations to improve representation robustness, particularly in low-data settings. These models leverage unlabeled molecular graphs to learn structure-aware embeddings and have demonstrated strong performance in downstream tasks such as property prediction, molecular similarity, and scaffold-based generalization.

In the 3D domain, recent models such as Uni-Mol [81] and Molformer3D [82] extend chemical language modeling to spatial representations by integrating 3D conformer modeling with sequence-level pretraining. Uni-Mol employs SE(3)-equivariant attention and is pre-trained on millions of molecular conformers through tasks like masked coordinate modeling and distance matrix reconstruction. Molformer3D combines transformer-based architectures with geometric encoding to capture invariant 3D molecular features. Geometry-enhanced methods such as GEM [83] further incorporate spatial information by modeling bond lengths, bond angles, and interatomic distances through a dual-graph architecture, enabling the learning of fine-grained geometric priors. EquiBind [84] introduces a geometric deep learning framework for direct ligand pose prediction in protein–ligand complexes, using SE(3)-equivariant networks to model atomic interactions without relying on exhaustive sampling. These models support a range of downstream applications, including conformation generation, molecular docking, and protein–ligand binding prediction. On the protein side, structure-aware language models have been developed to fuse AlphaFold2-derived spatial contexts with pretrained embeddings, allowing fine-tuning on structure-sensitive protein engineering tasks.

1.6 Aims of this Thesis

The objectives of this thesis are to investigate the use of Geometric Deep Learning (GDL) in three tasks central to drug discovery: drug–target interaction (DTI) prediction, molecular property prediction, and the discovery of novel inhibitors for challenging disease targets. This work aims to design effective GDL-based models to learn rich, multiscale representations from molecular data. The specific aims of this thesis are as follows:

Aim 1: Develop a bi-modal molecular property prediction model combining sequence- and graph-based representations.

Deep learning models exhibit different inductive biases. To leverage the complementary strengths of different modalities, we propose a unified bi-modal architecture that integrates representations from both SMILES sequences and molecular graphs, enabling more comprehensive and accurate property prediction.

Aim 2: Develop a dual-scale molecular representation framework to capture hierarchical and long-range structural dependencies.

To overcome the over-smoothing and over-squashing issues in conventional GNNs, we design a hierarchical model that combines GNN encoders and xLSTM modules at both the atom and motif levels, enabling more expressive and structurally aware molecular representations.

Aim 3: Develop a model to address target imbalance in molecular regression.

Molecular property datasets often exhibit skewed or long-tailed label distributions. We propose a distribution-aware expert routing framework that assigns samples to specialized experts based on their target value ranges, enabling more balanced learning across the label space.

Aim 4: Develop a DTI model capturing protein–ligand substructure interactions.

To enhance interpretability in drug–target interaction (DTI) prediction, we design an architecture that captures fine-grained interactions between molecular substructures and protein residues. An attention-inspired interaction module identifies key residue–substructure pairs, enabling precise binding inference and improved generalization across diverse targets.

Aim 5: Conduct a case study on PIN1 to demonstrate AI-guided virtual screening.

We conduct a virtual screening case study targeting PIN1, a key oncogenic driver implicated in triple-negative breast cancer. Leveraging our proposed molecular representation and routing models, we score a large compound library, filter candidates based on structural and interaction criteria, and prioritize promising inhibitors for potential therapeutic development.

1.7 Thesis Format and Publications

Following the research aims described above, this thesis adopts a manuscript-based format. It consists of four research articles and one case study, each addressing a specific aspect of geometric deep learning (GDL) for drug discovery. These works are presented as standalone but thematically connected chapters, aligning with the five aims of the thesis.

- **Molecular Property Prediction based on Bimodal Supervised Contrastive Learning**

This work introduces a general-purpose loss function that improves the embedding space for both classification and regression tasks. This design is later reused in other chapters to optimize structural representations. It has been published in the *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2022 [85]. **Included as Chapter 2.**

- **MolGraph-xLSTM: A graph-based dual-level xLSTM framework with multi-head mixture-of-experts for enhanced molecular representation and interpretability**

This work addresses limitations in the ability of GNNs to capture long-range dependencies, thereby complementing and extending the representation capacity initiated in Chapter 2. The manuscript has been submitted to *Communications Chemistry* (Manuscript ID: COMMSCHEM-25-0229A-Z) and is currently under peer review.

Included as Chapter 3.

- **DistRouting: Distribution-Aware Expert Routing Guided by Physicochemical Descriptors for Imbalanced Molecular Property Regression**

This work focuses on data imbalance in regression settings, further addressing regression tasks where previous framework-based improvements yielded limited gains in the rare regions. The manuscript has been submitted to *NeurIPS 2025* (Submission ID:17299) and is currently under peer review. **Included as Chapter 4.**

- **iNGNN-DTI: Prediction of Drug - Target Interaction with Interpretable Nested Graph Neural Network and Pretrained Molecule Models**

This work introduces a tailored method designed specifically for modeling interactions between small molecules and proteins. It has been published in *Bioinformatics, 2024* [86]. **Included as Chapter 5.**

- **Case Study: Discovery of PIN1 Inhibitors**

This work presents a case study on PIN1 inhibitor discovery, demonstrating how the methods developed in earlier chapters can be applied to a real-world drug discovery scenario. **Included as Chapter 6.**

Chapter 2

Molecular Property Prediction based on Bimodal Supervised Contrastive Learning

Yan Sun, Mohaiminul Islam, Ehsan Zahedi, Méline Kuenemann, Hassan Chouaib, Pingzhao Hu

Proceedings of the 2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Dec. 2022, pp. 394–397.

DOI: <https://doi.org/10.1109/BIBM55620.2022.9995689>

Reprinted with permission from IEEE.

AUTHORS CONTRIBUTIONS

Conceptualization: Y.S., E.Z., H.C., P.H.; Methodology: Y.S., M.I., E.Z., H.C., P.H.; Investigation: Y.S.; Writing original draft preparation: Y.S.; Writing review and editing: Y.S., M.I., E.Z., M.K., H.C., P.H.; Visualization: Y.S.; Supervision: E.Z., H.C., P.H. All authors have read and agreed to the published version of the manuscript.

2.1 Introduction

The prediction of molecular properties is an essential step in the discovery of new drugs, which saves time and cost by filtering out molecules with adverse properties in the early stage [87]. In recent years, many studies have explored new computational methods, especially deep learning-based methods [88], to handle the challenging task because they can significantly speed up the prediction of molecular properties.

A molecule can be represented in different formats, including the SMILES string and the molecular graph. SMILES describes molecular structures using ASCII string, and the molecular graph maps atoms and chemical bonds as vertices and edges, respectively. Many deep learning methods are designed to learn molecule representations. Graph Convolutional Networks (GCNs) have been widely used for molecular graph representation learning [61, 89, 90], and successful models in nature language processing (NLP) have been used to encode the SMILES string. The pretrained Bidirectional Encoder Representations from Transformers (BERT) models on SMILES strings may learn the information of fundamental properties of the molecule [74], while the representations learned by GCNs on the molecular graph may focus more on local structures. They could benefit property prediction in different ways. Intuitively, we consider using both of them.

Although we could have richer information when using more data, it is challenging to learn representations from multiple modalities due to information redundancy and distribution shift between modalities. In recent years, contrastive learning methods have been used in the multi-modal network to assist in aligning the representations and bridging the gap between different modalities[91, 92, 93, 94]. In this work, we propose a bimodal supervised contrastive learning (BSCL) framework to integrate the molecular graph and the SMILES string in a unified network. Inspired by [95], we designed a network which maps each modality to two subspaces: joint embedding space and modality-specific space. We optimize the joint embedding space with supervised contrastive loss. Supervised contrastive loss is expected to learn modality-invariant features with better boundaries among different classes. Furthermore, when we cal-

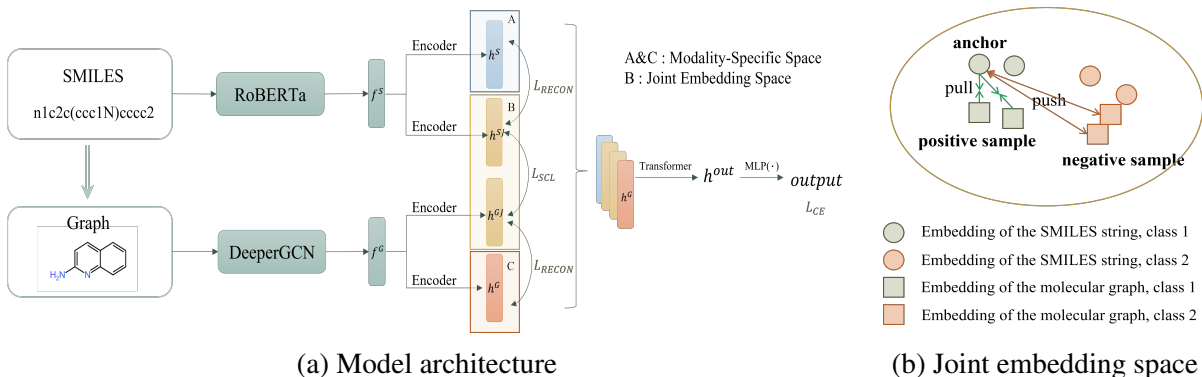


Figure 2.1: (a) **Model architecture.** The model has two branches: one for the SMILES string, and the other for the molecular graph. The SMILES string and the molecular graph are first encoded by RoBERTa and DeeperGCN, separately. Then, each of them is embedded in the joint embedding space and the modality-specific space. In the end, all embeddings are passed to a transformer module. The output of the transformer is concatenated as the feature of the input molecule. The prediction result is generated by a multilayer perceptron (MLP) module. (b) **Joint embedding space.** In the joint embedding space, the embedding of the SMILES string and the molecular graph is optimized by a supervised contrastive loss, which pulls positive samples closer to the anchor than negative samples.

calculate the supervised contrastive loss, we will set two samples (e.g. molecules) with the same label as a positive pair; otherwise, we will set them as a negative pair. However, in the regression datasets, there could be very few positive pairs because not many samples have the exact same label. A solution is to set two samples with a similar label as a positive pair. However, it may lose the details and therefore we propose a weighted supervised contrastive loss to help the model keep the difference between the similar samples.

The main contributions of this work are as follows: (1) We propose a bimodal architecture that integrates both the SMILES string and the molecular graph into a unified network with supervised contrastive learning. Experimental results demonstrate that this integration outperforms using either modality alone. (2) We introduce a weighted supervised contrastive loss designed for regression tasks, enabling more effective representation learning with continuous labels.

2.2 Related Work

Contrastive Learning for Multimodal Representation: Contrastive learning has proven effective in aligning representations across modalities by pulling together semantically similar samples and pushing apart dissimilar ones. It preserves both inter- and intra-modality relationships. Prior works have explored its application to multimodal tasks. For example, Udandaraao et al. [91] jointly learned image and text representations using contrastive loss, achieving strong performance on multimodal benchmarks. Yuan et al. [94] aligned images with their captions and demonstrated transferability to various vision tasks.

Contrastive Learning for Molecular Representation: Self-supervised contrastive learning has been applied to molecular graphs, where positive pairs are typically generated via data augmentation strategies such as node/edge perturbation [80, 96]. However, such augmentations may compromise chemical integrity. Supervised contrastive learning (SCL) [97] addresses this issue by utilizing label information, avoiding destructive augmentations and enabling more semantically grounded representation learning.

Multimodal Learning for Molecules: Recent studies have explored integrating multiple molecular modalities. Dual-view Molecule Pre-training (DMP) [98] jointly models SMILES strings and molecular graphs using a dual-view consistency loss. Similarly, GeoMGCL [99] aligns 2D and 3D molecular graphs using a self-supervised contrastive loss. These approaches demonstrate the benefit of leveraging complementary modalities in molecular learning.

2.3 The Proposed BSCL Method

In this section, we discuss the details of our BSCL method. We first formulate the problem in subsection A. Then, we introduce the architecture of the model and the loss function in subsection B and C, respectively. In subsection D, we discuss the way to combine the global molecular feature with the representation learned by our deep learning model.

2.3.1 Problem Formulation

For a given molecule, it can be represented as a SMILES sequence S or a molecular graph $G = (V, E)$, and the properties of the molecule are labeled as $y = n_1, n_2, \dots, n_T$, T is the number of properties. $n \in \{0, 1\}$ when the dataset is a classification task, and $n \in \mathbb{R}$ when the dataset is a regression task.

2.3.2 Model Architecture

As shown in Fig.2.1a, there are two branches in our model. The SMILES string S is encoded by a pretrained RoBERTa [77] model as F_S , and the molecular graph G is encoded by Deep-erGCN [90] as F_G . $F_S, F_G \in \mathbb{R}^d$, d is the dimension of the feature vector. F_S and F_G are mapped to two domains separately, and we can get four representations, H_S, H_G, H_{SJ}, H_{GJ} , of the molecule, among which H_{SJ}, H_{GJ} are in the joint embedding space and H_S, H_G are in the modality-specific space. In the end, all representations are stacked and passed to a Transformer module. The output of the Transformer is the final representation of the molecule, and we pass it to a multilayer perceptron (MLP) for the prediction.

2.3.3 Loss Function

The loss consists of three parts: supervised contrastive loss, reconstruction loss, and task loss. The loss is calculated in the following form:

$$L = L_{task} + w_{SCL}L_{SCL} + w_{recon}L_{recon} \quad (2.1)$$

w_{SCL} and w_{recon} are weights for the supervised contrastive loss and the reconstruction loss.

Supervised Contrastive Loss

The joint embedding space is optimized by supervised contrastive loss [97]. To be specific, given a batch of N molecules, the embedding of the molecule with index $i \in I \equiv \{1, 2, \dots, N\}$ is

represented as h_i^{SJ} for the SMILES string and as h_i^{GJ} for the molecular graph in the joint embedding space. Both h_i^{SJ} and h_i^{GJ} are mapped with a MLP to obtain the vectors $z_i^S = MLP(h_i^{SJ})$ and $z_i^G = MLP(h_i^{GJ})$, where $z_i^S, z_i^G \in \mathbb{R}^d$. The supervised contrastive loss is calculated with z^S and z^G . We construct sample pairs $\{z^S, z^G\}$ between SMILES strings and molecular graphs. The objective is to maximize the similarity of positive pairs by contrasting them with negative pairs. We will introduce the loss function for classification tasks and regression tasks, respectively.

Supervised Contrastive Loss for Classification Tasks For a classification task, given an anchor molecule i , we define samples with the same label for the molecule i as its positive samples. In a batch of data, we can generate a set P^i that includes the index of positive samples for the molecule i . The supervised contrastive loss can be calculated in the following form:

$$\mathcal{L}^i = \frac{-1}{|P^i|} \sum_{p \in P^i} \left(\log \frac{\exp(z_i^S \cdot z_p^G / \tau)}{\sum_{k=1, k \neq i}^N \exp(z_i^S \cdot z_k^G / \tau)} + \log \frac{\exp(z_i^G \cdot z_p^S / \tau)}{\sum_{k=1, k \neq i}^N \exp(z_i^G \cdot z_k^S / \tau)} \right), \quad (2.2)$$

$$\mathcal{L}_{SCL} = \sum_{i=1}^N \mathcal{L}^i, \quad (2.3)$$

$\tau \in \mathcal{R}^+$ is a scalar hyperparameter.

If there are multiple tasks in one dataset, we calculate the loss for each task and average the loss of all tasks:

$$\mathcal{L}_{SCL} = \frac{1}{T} \sum_{t=1}^T \sum_{i=1}^N \mathcal{L}_t^i. \quad (2.4)$$

Supervised Contrastive Loss for Regression Tasks For a regression task, we first preprocess the label with z-score standardization. We calculate the mean μ and the standard deviation σ of the training set and normalize the label of all molecules with the following equation:

$$\hat{y}_i = \frac{y_i - \mu}{\sigma}. \quad (2.5)$$

Then, given an anchor molecule i , we calculate the Euclidean distance between the label of the anchor molecule i and other samples. We set samples with the distance to i less than the threshold l_i as positive samples; otherwise, the samples are negative samples; we represent the negative set of the molecule i by N^i . It is reasonable to assume that the closer the samples are to the anchor sample, the more important when they are positive samples, while less important when they are negative samples. Therefore, we set a weight for each sample based on their distance from the anchor molecule. The supervised contrastive loss is calculated as:

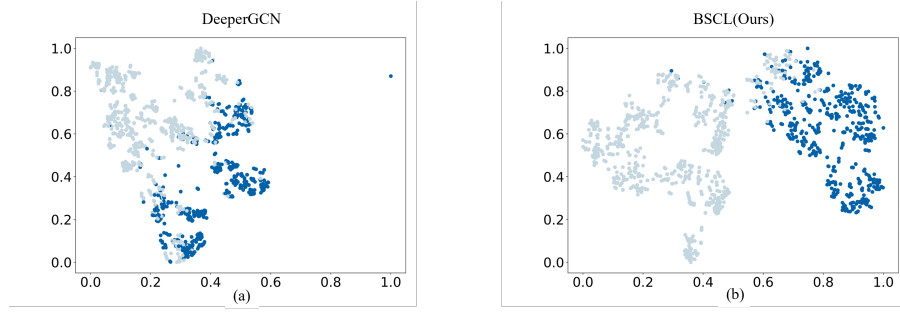


Figure 2.2: **t-SNE visualization of the embedding space of DeeperGCN and BSCL trained on the BACE dataset.** Points in different colors are of different classes.

$$\mathcal{L}^i = \frac{-1}{|P^i|} \sum_{p \in P^i} \left(\log \frac{\exp(w_{ip} z_i^S \cdot z_p^G / \tau)}{\frac{1}{|N^i|} \sum_{n \in N^i} w_{in} \exp(z_i^S \cdot z_n^G / \tau)} + \log \frac{\exp(w_{ip} z_i^G \cdot z_p^S / \tau)}{\frac{1}{|N^i|} \sum_{n \in N^i} w_{in} \exp(z_i^G \cdot z_n^S / \tau)} \right) \quad (2.6)$$

Let the distance between the molecule i and other samples be $D = \{d_{i1}, d_{i2}, \dots, d_{iN}\}$. The weights w_{ip} and w_{ik} are calculated as:

$$w_{ip} = 1 + \exp(-\gamma_1 * \frac{d_{ip}}{l_i}), \quad w_{in} = \exp(\frac{d_{in} - l_i}{\max(D) - l_i} + \gamma_2). \quad (2.7)$$

w_{ip} decreases with increasing d_{ip} , while w_{ik} reverses. l_i is setted as the median value of D . γ_1 and γ_2 are hyperparameters and adjust the rate of change in w_{ip} and w_{in} . $\max(\cdot)$ calculates the maximum value of D .

Reconstruction Loss

The reconstruction loss [95] is added to ensure that the features of each modality is representative. It is achieved by reconstructing f^S using h^S and h^{SJ} , and reconstructing f^G using h^G and h^{GJ} . To be specific, h^S and h^{SJ} are first concatenated as $[h^S \cdot h^{SJ}] \in \mathbb{R}^{2 \times d}$, and then we decode $[h^S \cdot h^{SJ}]$ as $\hat{f}^S \in \mathbb{R}^d$ which is the reconstruction vector of f^S . The reconstruction vector $\hat{f}^G \in \mathbb{R}^d$ is generated with the same operations. The loss can be calculated as follows:

$$L_{recon} = (\|f^S - \hat{f}^S\|_2^2 + \|f^G - \hat{f}^G\|_2^2)/2. \quad (2.8)$$

$\|\cdot\|_2^2$ is the squared L_2 - norm.

Task Loss

The task loss evaluates the difference between the output of the network and the label of the dataset. Representations from different domains are fused by a transformer module, and the outputs of the transformer are concatenated. The final representation of the molecule can be represented as:

$$h^{out} = \text{concat}(\text{Transformer}([h^S, h^G, h^{SJ}, h^{GJ}]]), \quad (2.9)$$

$$h^{out} \in \mathbb{R}^{4 \times d}.$$

The representation of a molecule is passed to a MLP module for prediction:

$$\text{output} = \text{MLP}(h^{out}). \quad (2.10)$$

We evaluate the difference between the output and the label with cross-entropy loss for the classification task, and mean squared error for the regression task:

$$\begin{aligned}
L_{task-classification} &= -\frac{1}{N_{task}} \sum_{i=0}^{N_{task}} y_i \cdot \log \widehat{y}_i, \\
L_{task-regression} &= \frac{1}{N_{task}} \sum_{i=0}^{N_{task}} \|y_i - \widehat{y}_i\|_2^2.
\end{aligned}
\tag{2.11}$$

2.3.4 Including Global Feature

Directed MPNN (D-MPNN) [61] proves that the combination of the molecular features learned by the deep learning model and the global molecular features can further improve the prediction performance. As mentioned in [61], some datasets are too small to learn identified features. In this paper, we also verify the performance with the global features. We use the molecular access system (MACCS) key [100] and the extended-connectivity fingerprint (ECFP) [101]. Both of them can be generated by the open source package RDKit [102]. We directly concatenate the two features as h^{global} . We find in practice that we need to concatenate h^{global} and h^{out} in different ways for the classification task and the regression task. For the classification task, we directly concatenate h^{global} and h^{out} , and pass it to a MLP module. For the regression task, we preprocess h^{out} with z-score normalization and h^{global} with $L_2 - norm$ before the concatenation.

Table 2.1: Dataset details.

Dataset	#Molecule	#Property	Task Type	Split
BACE	1522	1	Classification	Scaffold
ClinTox	1491	2	Classification	Random
Tox21	8014	12	Classification	Random
Sider	1427	27	Classification	Random
Freesolv	643	1	Regression	Random
ESOL	1128	1	Regression	Random
Lipophilicity	4200	1	Regression	Random

Table 2.2: Hyperparameters for each dataset.

Dataset	w_{scl}/w_{recon}	Learning Rate	Epoch	γ_1/γ_2
BACE	1/0.1	0.0001	300	- ^a
ClinTox	1/0.1	0.0001	300	-
Tox21	1/0.1	0.0001	300	-
Sider	1/0.1	0.0001	300	-
Freesolv	1/1	0.001	800	2/1
ESOL	1/1	0.001	300	2/2
Lipophilicity	1/1	0.001	600	4/3

^a γ_1 and γ_2 are only for regression tasks.

Table 2.3: Performance for regression tasks and classification tasks.

Dataset	DeeperGCN	ChemBERTa	D-MPNN	GEM	BSCL	BSCL+Global Feature
Regression (RMSE ↓)						
Freesolv	1.153±0.045 ^a	1.528±0.032	1.075±0.054	1.141±0.065	1.044±0.138	1.084±0.067
ESOL	0.566±0.068	0.719±0.061	0.555±0.047	0.554±0.048	0.548±0.027	0.542±0.027
Lipophilicity	0.555±0.021	0.693±0.029	0.555±0.023	0.551±0.026	0.547±0.024	0.563±0.018
Classification (AUROC ↑)						
BACE	0.801±0.006	0.771±0.003	- ^b	0.856±0.011	0.826±0.015	0.833±0.009
ClinTox	0.844±0.027	0.943±0.019	0.864±0.017	0.862±0.026	0.962±0.008	0.975±0.008
Tox21	0.840±0.002	0.751±0.011	0.851±0.002	0.856±0.007	0.850±0.007	0.857±0.011
Sider	0.644±0.014	0.622±0.026	0.676±0.014	0.678±0.021	0.663±0.017	0.696±0.008

^a The result is the mean and the standard deviation of three independent runs.

^b D-MPNN results come from [103], and the result of the BACE dataset is not provided by the paper.

2.4 Experiments

2.4.1 Dataset and Evaluation

We evaluate the performance on seven datasets from MoleculeNet [36] and the details of the datasets are shown in Table 2.1. Each dataset is split into train, validation and test subsets by the proportion of 80%:10%:10%, respectively. Scaffold splitting splits the dataset based on the two-dimensional structure of the samples, so the samples in the test and validation subsets will be more distinguished from the samples in the training subset. We use the data provided by [103] which has split into the three subsets. We save the model with the best performance in the validation set during training and evaluate it with the test set. Three independent runs are performed on each dataset and we calculate the mean and the standard deviation. The evaluation metric is the root-mean-square error (RMSE) for the regression tasks and the area under curve of the receiver operating characteristics (AUROC) for the classification tasks.

2.4.2 Baselines

We compare our work with 4 methods: DeeperGCN, ChemBERTa, D-MPNN, and GEM. The results of D-MPNN come from [103] while the others are generated by ourselves. Brief descriptions of the baselines are as follows:

- **DeeperGCN** [90] adapts residual connections, dilated connections, and also dense connections to GCNs which is helpful in solving the gradient vanishing problem in GCNs and makes DeeperGCN a very deep network.
- **ChemBERTa** [77] trains RoBERTa with large molecule datasets. In this work, we use the model pre-trained with 10M molecules from ZINC dataset [104].

- **Directed MPNN (D-MPNN)** [61] is a graph neural network that uses messages associated with directed edges.
- **Geometry-enhanced molecular representation learning method (GEM)** [105] designs a geometry-based graph neural network and also various geometry-level self-supervised tasks to learn the coarse three-dimensional structures.

2.4.3 Parameter Setting

To compare with DeeperGCN, we first tune the hyperparameters of DeeperGCN and use the same network architecture as our network. The number of layers in DeeperGCN is 3 for regression tasks and 7 for classification tasks. The dimension of the two embedding spaces is 128. The network is optimized with Adam optimizer. The detailed setting for each dataset can be found in Table 2.2.

2.4.4 Result and analysis

Effectiveness of the bimodal architecture

Our bimodal network uses DeeperGCN for the encoding of the molecular graph and ChemBERTa for the encoding of the SMILES string. We compare our performance with the result using DeeperGCN or ChemBERTa alone to investigate the effectiveness of the fusion. As shown in Table 2.3, we could get an improvement on all seven datasets. The largest improvement is 3.1% on classification tasks and 9.5% on regression tasks compared to the best performance between DeeperGCN and ChemBERTa. In Fig. 2.2, we visualize the features learned by DeeperGCN and our method on the BACE dataset. We observe that the features learned by our method on the BACE dataset is more separable.

Comparison to other methods using single modality

When comparing to the two cutting-edge methods, D-MPNN and GEM, our method performs better on the regression tasks than on the classification tasks. As shown in Table 2.3, both DeeperGCN and ChemBERTa are worse than D-MPNN and GEM on all datasets except for ClinTox. After fusion using our method, we can achieve the best performance on all regression datasets. For the classification tasks without using global features, the improved performance still does not surpass the state-of-the-art. However, we find that the combination of the global feature and the representation learned by our model can further improve our results in the classification datasets with the highest improvement at 4.9% in the Sider dataset.

Analysis of the Weighted Supervised Contrastive Loss

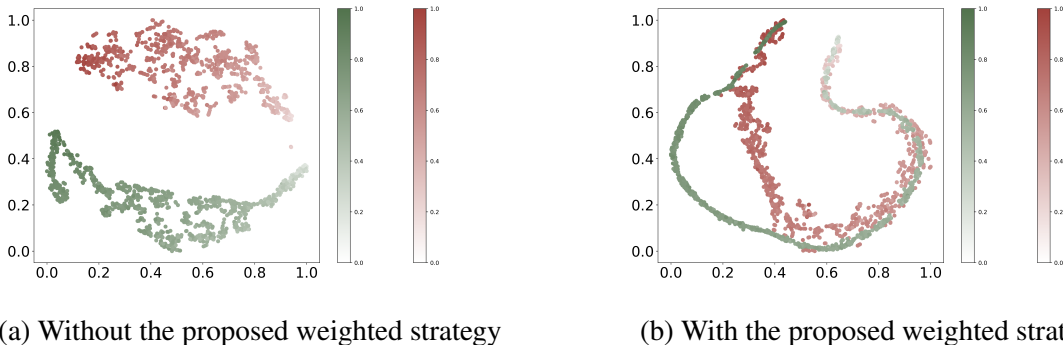


Figure 2.3: t-SNE visualization of the joint embedding space on the ESOL dataset. The label of each sample is normalized to a range of 0 to 1, where 0 is the minimum value and 1 is the maximum value. Mappings in green are graph samples, and in red are SMILES samples. When using the proposed weighted strategy, the joint embedding space shows continuity corresponding to the continuity in the target space.

Effectiveness of the weighted supervised contrastive loss If we use the vanilla supervised contrastive loss (Equation 2.2) in regression tasks, the RMSE of the ESOL dataset is 0.566 ± 0.035 while the result is 0.548 ± 0.027 when using the proposed loss (Equation 2.6). We assume that using the proposed loss function is helpful to maintain the relationship among samples. We visualize representations in the joint embedding space with t-SNE, and the result is shown in

Fig.2.3. It can be seen that when we use the proposed weights, samples with a similar label are also close to each other in the embedding space, which may benefit the prediction performance.

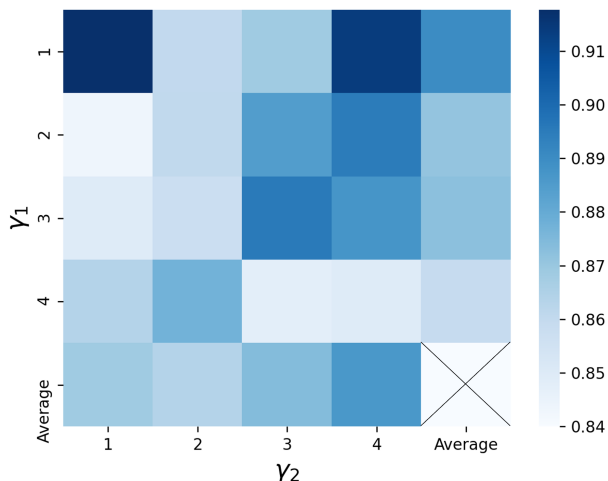


Figure 2.4: **RMSE as a function of γ_1 and γ_2 .** We sweep different combinations of γ_1 and γ_2 , and the color illustrates the RMSE value (the lighter, the better). The result is generated in the Freesolv validation dataset.

Hyperparameter analysis In this section, we investigate the influence of the hyperparameters γ_1 and γ_2 . We train the model with different combinations of γ_1 and γ_2 and record the RMSE of the validation set. The experiment was carried out on the Freesolv dataset. We show the result with a heatmap in which each cell corresponds to a pair of combinations, and the color represents the RMSE value (the lighter the better). We also calculate the average value when we fix one parameter and change another parameter. As Fig.2.4 shows, the average RMSE decreases with increasing γ_1 , and increases with increasing γ_2 . Therefore, in this work, we apply a grid search to the regression tasks with values of γ_1 in [2,3,4] and values of γ_2 in [1,2,3]. Furthermore, we also check the performance on the test set, and the largest RMSE among all combinations is 1.120 ± 0.081 which is still smaller than DeeperGCN (1.153 ± 0.045) and ChemBERTa (1.528 ± 0.032). Therefore, our method is not that sensitive to the selection of γ_1 and γ_2 .

Ablation Study

Table 2.4: **Ablation test on the BACE and Freesolv datasets.** Abbreviations: CE = Cross Entropy, SCL = Supervised Contrastive Loss, Recon = Reconstruction Loss.

Test	BACE (AUROC $\uparrow$)	Freesolv (RMSE $\downarrow$)
CE	0.797 ± 0.015	1.142 ± 0.056
CE + SCL	0.807 ± 0.008	1.070 ± 0.139
CE + SCL + Recon	0.826 ± 0.015	1.044 ± 0.138

Loss function In our proposed method, we optimize the model with three losses. This section is to investigate the effectiveness of each loss. We evaluated it on the BACE dataset and the Freesolv dataset. We first generate the result using only the cross-entropy loss, and then add each loss one by one. Results in Table 2.4 show an improvement of 1.2% and 6.3% in the BACE and the Freesolv dataset, respectively, when we add the supervised contrastive loss.

The necessity of the specific domain To prove the need for a specific domain, we remove the specific domain and test the performance in the BACE and the Freesolv dataset. The test results are 0.810 ± 0.008 and 1.11 ± 0.09 for the BACE and the Freesolve, respectively, which are worse than their performance (0.826 ± 0.015 and 1.044 ± 0.138) with the specific domain.

2.5 Summary

In this work, we proposed a novel bimodal supervised contrastive learning (BSCL) method for molecular property prediction. Our approach aimed to integrate the complementary information encoded in the SMILES string and the molecular graph within a unified network architecture. We introduced supervised contrastive learning to optimize the molecular embedding space and further designed a weighted supervised contrastive loss tailored for regression tasks. Experimental results demonstrated the effectiveness of the proposed method. The results were

reported as mean $\pm$ standard deviation across multiple runs, which provides a balanced view of both predictive accuracy and variability; although the variance is not always the lowest, the consistently superior average performance indicates the reliability of our approach. One limitation is that the loss design can be sensitive to hyperparameter settings, suggesting that more adaptive or automated optimization strategies may further enhance its robustness.

Chapter 3

MolGraph-xLSTM: A graph-based dual-level xLSTM framework with multi-head mixture-of-experts for enhanced molecular representation and interpretability

Yan Sun, Yutong Lu, Yan Yi Li, Zihao Jing, Carson K. Leung, Pingzhao Hu

arXiv:2501.18439 DOI: <https://doi.org/10.48550/arXiv.2501.18439>

Under review at Communications Chemistry (Manuscript ID: COMMSCHEM-25-0229A-Z).

AUTHORS CONTRIBUTIONS

Conceptualization: Y.S., Y.L., Y.Y.L., Z.J., P.H.; Methodology: Y.S., P.H.; Investigation: Y.S., Y.L., Y.Y.L., Z.J., P.H.; Data curation: Y.S., Y.L., Y.Y.L.; Formal analysis: Y.S.; Software: Y.S.; Visualization: Y.S.; Writing original draft preparation: Y.S.; Writing review and editing: Y.S., Y.L., Y.Y.L., Z.J., P.H., C.L.; Supervision: P.H., C.L.; Funding acquisition: P.H. All authors have read and agreed to the published version of the manuscript.

3.1 Introduction

Predicting the molecular properties of a compound, particularly its ADMET (Absorption, Distribution, Metabolism, Excretion, and Toxicity) characteristics, is critical during the early stages of drug development [106, 107]. Leveraging deep learning for molecular representation to predict their properties significantly enhances the efficiency of identifying potential drug candidates [108, 109]. Molecular graphs retain richer structural information, which is crucial for accurate property prediction. In recent years, graph neural networks (GNNs) built on molecular graph data have been extensively utilized for molecular representation learning to predict their properties [61, 79, 80, 83, 90, 110, 111, 112, 113].

A key challenge in molecular property prediction lies in capturing long-range dependencies—the influence of distant atoms or substructures within a molecule on a target property. While graph neural networks (GNNs) leverage neighborhood aggregation as their core mechanism—updating the hidden states of each node by aggregating information from neighboring nodes using operations like sum, max, or mean pooling [114, 115]—they face significant limitations in capturing these long-range dependencies. Specifically, over-smoothing and over-squashing hinder their performance. Over-smoothing occurs when, as the number of layers increases, node representations become increasingly similar, leading to a loss of distinction between nodes [116]. On the other hand, over-squashing refers to the compression of information from distant nodes as it propagates toward the target node, making it challenging for relevant information to be effectively transmitted [117]. These issues limit the ability of GNNs to fully exploit global structural information, reducing their effectiveness in complex molecular property prediction tasks.

To address these challenges, we propose the MolGraph-xLSTM model, which integrates the xLSTM architecture with molecular graphs. Traditionally, Long Short-Term Memory (LSTM) networks have been widely used in natural language processing (NLP) tasks to capture sequential data representations [118]. With its gating mechanisms, LSTM effectively decides which information to retain or discard, enabling it to manage long-range dependencies. Thus,

we incorporate LSTM into our model to address the limitations of GNNs in handling long-range information. Recently, an improved version, xLSTM, was introduced [119]. xLSTM includes two additional modules, sLSTM and mLSTM, which expand the storage capacity of the original LSTM. Experimental results have shown favorable performance compared to two state-of-the-art architectures: Transformer [120] and State Space Models [121]. For this reason, we choose this enhanced LSTM model in our framework.

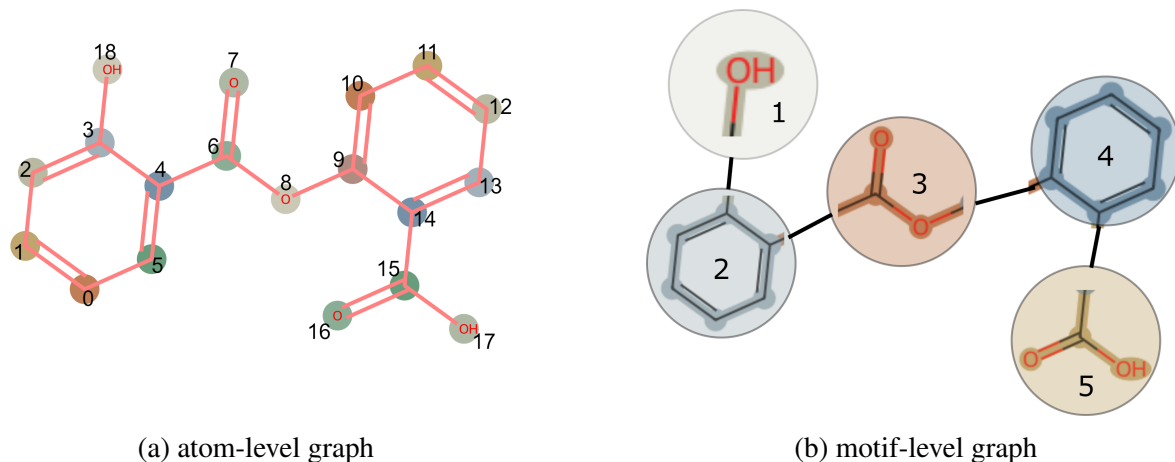


Figure 3.1: **Comparison of atom-level and motif-level graph representations.** (a) The atom-level graph represents each atom as a node and each bond as an edge. (b) The motif-level graph combines substructures into single nodes, resulting in a graph that is less complex than the atom-level graph.

We utilize both atom-level and motif-level molecular graphs in our approach (Figure 3.1). In the atom-level graph, each node represents an atom and each edge represents a bond within the molecule. The motif-level graph, on the other hand, is a partitioned version of the atom-level graph, where each node represents a substructure (such as aromatic ring) within a molecule. This results in a significantly simplified representation compared to the atom-level graph. Such simplification aids the model in learning features linked to local structures, as similar local motifs, from a functional group perspective, tend to impart similar properties to molecules [122]. Furthermore, the simplified motif-level graph, by reducing complexity and eliminating cycle structures, becomes more closely with sequential data. This structural simplification aligns well with the strengths of xLSTM, which is inherently designed to handle sequential informa-

tion, making the motif-level graph more suitable for processing with xLSTM.

However, relying solely on the motif-level graph would not capture all molecular details effectively, and motif partitioning itself demands precise segmentation. Therefore, we incorporate both atom-level and motif-level graphs in our model. For the atom-level representation, we introduce a GNN-based xLSTM with jumping knowledge [123]. Here, the GNN collects local information from the atom-level graph, and jumping knowledge aggregates features from multiple GNN layers, producing enriched node representations as inputs to xLSTM. By combining features from both the atom- and motif-level graphs, we constructed a comprehensive molecular representation for accurate property prediction.

Additionally, we integrate the multi-head mixture-of-experts (MHMoE) module [124] to enhance the predictive performance of our model. The sparse mixture-of-experts (SMoE) [125] framework has been demonstrated as an effective method for scaling models while maintaining computational efficiency by dynamically assigning inputs to different expert networks. This allows the input features to be processed by multiple experts, enabling diverse perspectives and improving the quality of learned representations. Building upon SMoE, the MHMoE architecture introduces further advancements by enhancing the usage of expert and promoting a more fine-grained understanding of input features. By incorporating the MHMoE module, our model is able to generate more expressive feature representations, which enhances its predictive accuracy.

The contributions of our work are as follows:

- **Adaptation of xLSTM to dual-level molecular graph representation:** We design a unified architecture that applies the extended LSTM (xLSTM) to both atom-level and motif-level molecular graphs. At the atom level, xLSTM follows GCN layers to enhance local features with long-range context. At the motif level, the graph is simplified through functional substructure decomposition, resulting in a sequential-like topology that further aligns with xLSTM’s modeling strengths. This dual-level application enables comprehensive capture of fine-grained and high-level structural dependencies, substantially

boosting prediction performance across 21 molecular property benchmarks.

- **Integration of Multi-Head Mixture-of-Experts (MHMoE) for enhanced prediction:**

We incorporated the multi-head mixture-of-experts (MHMoE) module into our framework, which dynamically assigns input features to different expert networks, enabling diverse feature processing and improving predictive accuracy. This architecture refines feature representations through fine-grained expert activation.

- **Case study analysis for model interpretability:** We conducted case study to investigate the substructures assigned the highest weights by the network, demonstrating that the atom-level and motif-level information are complementary. By cross-referencing with known literature, we identified strong correlations between the highlighted substructures and specific molecular properties, underscoring the ability of the model to implicitly learn biologically relevant information.

3.2 Background and Related Work

3.2.1 Extended Long Short-Term Memory

Long Short-Term Memory (LSTM) networks are designed to process sequence data by incorporating a memory cell regulated by three gates: the input gate i_t , output gate o_t , and forget gate f_t . The input gate controls how much new information is added to the memory, the forget gate decides how much of the past information to retain, and the output gate determines what part of the memory contributes to the current hidden state. At each time step t , the memory cell is updated by combining the retained memory from the previous step with new information, ensuring the network can selectively remember or forget information as needed. The update of

the memory cell can be represented as:

$$c_t = f_t \odot c_{t-1} + i_t \odot z_t, \quad (3.1)$$

$$h_t = o_t \odot \psi(c_t), \quad (3.2)$$

$$z_t = \varphi(w_z^\top x_t + r_z h_{t-1} + b_z), \quad (3.3)$$

$$i_t = \sigma(w_i^\top x_t + r_i h_{t-1} + b_i), \quad (3.4)$$

$$f_t = \sigma(w_f^\top x_t + r_f h_{t-1} + b_f), \quad (3.5)$$

$$o_t = \sigma(w_o^\top x_t + r_o h_{t-1} + b_o), \quad (3.6)$$

where c_t denotes the cell state, h_t represents the hidden state and z_t represents the candidate state. The terms w_z , w_i , w_f , and w_o are weight vectors associated with the input vector x_t , while r_z , r_i , r_f , and r_o are weight vectors corresponding to the previous hidden state h_{t-1} . The bias terms are given by b_z , b_i , b_f , and b_o . The functions $\psi(\cdot)$, $\varphi(\cdot)$, and $\sigma(\cdot)$ are activation functions, where $\psi(\cdot)$ and $\varphi(\cdot)$ typically represent *tanh* function, and $\sigma(\cdot)$ denotes the *sigmoid* function.

In xLSTM, two new blocks are introduced: sLSTM and mLSTM. Both incorporate exponential gating to enhance the memory cells of the original LSTM. The function $\sigma(\cdot)$ in equations 3.4 and 3.5 is modified to use exponential activation. mLSTM further extends the memory capacity by introducing a matrix memory, which improves the storage capabilities of LSTM. Specifically, it replaces the scalar cell state c in equation 3.1 with a matrix. The xLSTM block is constructed by stacking alternating sLSTM and mLSTM blocks, while the full xLSTM architecture consists of multiple xLSTM blocks stacked together.

3.2.2 Deep Learning Methods Based on Molecular Graphs for representation learning to predict molecular properties

GNN is the most widely used architecture for molecular graph representation. GNN updates node features through a two-step process: first, aggregating information from neighboring nodes, followed by applying a Multi-Layer Perceptron (MLP) to update the own features. Various GNN-based models have been proposed for molecular property prediction. The Directed-Message Passing Neural Network (DMPNN) [61] optimizes message passing by centering aggregation on bonds rather than atoms, effectively avoiding redundant loops. DeeperGCN [90] focuses on training very deep GCNs by introducing an improved residual connection to enhance performance. The Hierarchical Informative Graph Neural Network (HiGNN) [126] segments the molecular graph into fragments using Breaking of Retrosynthetically Interesting Chemical Substructures (BRICS) [127]. Both the original molecular graph and its fragments are then processed by the GNN to generate a hierarchical representation of the molecule. FP-GNN [111] integrates molecular fingerprints with Graph Attention Networks (GAT), combining traditional descriptors with GNN features to improve prediction accuracy. These approaches primarily rely on GNNs for feature extraction. While DeeperGCN and DMPNN focus on optimizing message-passing mechanisms, HiGNN introduces hierarchical information through fragment-based graph segmentation, and FP-GNN enhances GNN-based features by incorporating traditional molecular fingerprints. Beyond GNN-exclusive models, hybrid architectures have been proposed. For example, TransFoxMol [112] designed an embedding unit that combines a GNN and a transformer to balance the local and long-range interactions of an atom.

3.2.3 Multi-Head Mixture of Experts

The Mixture of Experts (MoE) is a classical ensemble method that combines multiple experts with identical architectures, routing inputs to specific experts via a gating mechanism

[128, 129]. This design enables different experts to specialize in processing distinct types of information.

Recently, the application of MoE in deep learning has gained significant attention. The Sparsely-Gated Mixture-of-Experts (SMoE) layer was introduced, where each input is routed to the top- K ($K \geq 2$) most appropriate experts [125]. Building upon this, the Multi-Head Mixture-of-Experts (MHMoE) [124] was proposed, which partitions the input into multiple segments, enabling each segment to be processed by the top- K selected experts.

Consider a system with n experts, denoted as $E_1, E_2, \dots, E_n$, and an input $x \in \mathbb{R}^{h \times d}$. The input x is divided into h segments, $x_1, x_2, \dots, x_h$, where each segment is d -dimensional. The output of the MoE layer for a given segment x_s is calculated as:

$$x_s^{\text{MoE}} = \sum_{e=1}^n G(x_s)_e E_e(x_s), \quad (3.7)$$

where $G(x_s)_e$ is the gating function that assigns a weight to the i -th expert. The gating function $G(x_s)_e$ is defined as:

$$G(x_s) = \text{softmax}(\text{TopK}(g(x_s) + D_{\text{noise}}, K)), \quad (3.8)$$

with $g(x_s)$ computing the raw scores for each expert and D_{noise} adding stability and exploration during training. The TopK function filters the top- K elements of a vector v as follows:

$$\text{TopK}(v, K)_s = \begin{cases} v_s & \text{if } v_s \text{ is among the top } K \text{ elements of } v, \\ -\infty & \text{otherwise.} \end{cases} \quad (3.9)$$

Finally, the MHMoE output is obtained by concatenating the outputs of all segments pro-

cessed by the MoE layer:

$$x^{\text{MHMoE}} = \text{CONCAT}(x_1^{\text{MoE}}, x_2^{\text{MoE}}, \dots, x_h^{\text{MoE}}). \quad (3.10)$$

3.3 Method

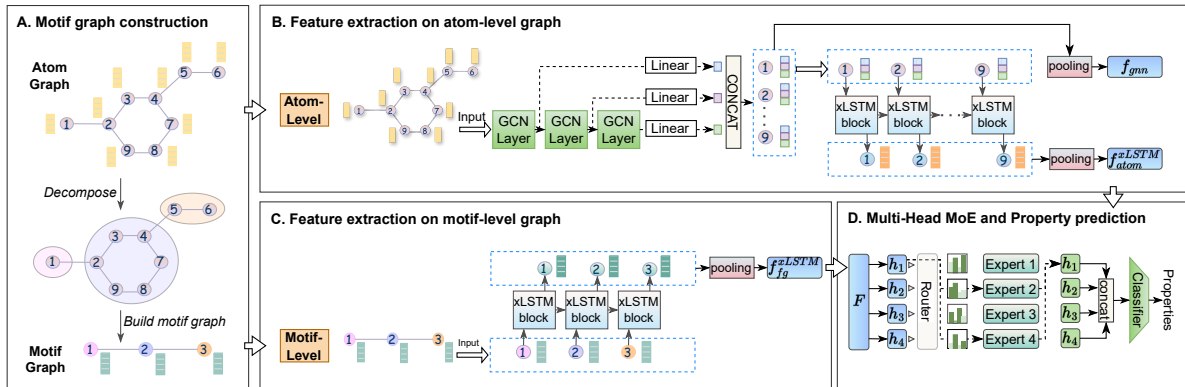


Figure 3.2: **Architecture of MolGraph-xLSTM.** The architecture consists of four main components: (A) Motif graph construction. The atom-level graph is decomposed into motifs to form a motif-level graph. (B) Feature extraction on the atom-level graph. A GCN-based xLSTM framework with jumping knowledge extracts features, followed by pooling to generate the atom-level representation f_{atom}^{xLSTM} . (C) Feature extraction on the motif-level graph. Using xLSTM blocks and pooling to produce the motif-level representation f_{motif}^{xLSTM} . (D) Multi-Head Mixture-of-Experts (MHMoE) and property prediction. Features (f_{gcn} , f_{atom}^{xLSTM} and f_{motif}^{xLSTM}) are combined and refined through the MHMoE module for final property prediction.

3.3.1 Construction of Atom- and Motif-Level Molecular Graphs

Starting from the SMILES string of a molecule, we convert it into an atom-level molecular graph $G_{\text{atom}} = \{V_{\text{atom}}, E_{\text{atom}}\}$ using RDKit tool [102], where $V_{\text{atom}} = \{v_p^{\text{atom}}\}$ represents the set of nodes, and $E_{\text{atom}} = \{(v_p^{\text{atom}}, v_q^{\text{atom}})\}$ represents the set of edges. Each node v_p^{atom} corresponds to an atom and is initialized with 11 atomic features, including atomic number, chirality, and aromaticity (Table 3.1). Likewise, each edge $(v_p^{\text{atom}}, v_q^{\text{atom}})$ represents a bond and includes features such as bond type, stereochemistry, and conjugation (Table 3.2). Based on the atom-level

Table 3.1: Atom features and descriptions.

Feature Type	Description	Feature Size
Atom Element	The chemical symbol of the atom (e.g., C, N, O, S, etc.).	44
Atom Degree	The number of directly bonded neighboring atoms.	11
# Attached Hydrogens	The total number of hydrogen atoms directly bonded to the atom.	11
Implicit Valence	The implicit valence of the atom.	11
Total Valence	The total valence of the atom.	11
Formal Charge	The formal charge of the atom.	11
Hybridization State	The hybridization state of the atom.	6
Radical Electrons	The number of unpaired electrons associated with the atom.	6
Chirality	The chirality of the atom.	5
Aromaticity	Indicates whether the atom is aromatic.	1
Ring Membership	Indicates whether the atom is part of a cyclic structure.	1

Table 3.2: Bond features and descriptions.

Feature Type	Description	Feature Size
Bond Type	The type of the bond between two atoms (e.g., single, double, triple, or aromatic).	4
Bond Stereochemistry	The stereochemistry of the bond.	6
Conjugation	Indicates whether the bond is part of a conjugated system.	1

graph, we then generate a motif-level graph $G_{motif} = \{V_{motif}, E_{motif}\}$ through the ReLMole, as described by [130]. In ReLMole, three types of substructures are considered as motifs: rings, non-cyclic functional groups, and carbon-carbon single bonds. In this motif graph, each node represents a motif and is initialized with 12 features. Each edge represents the connection between two motifs. Details of the initial features are provided in Table 3.3.

Both node and edge features are embedded into a d -dimensional feature vector. Specifically, we denote the input node feature of the atom-level graph and the motif-level graph as $H_{atom}^0 \in \mathbb{R}^{N_{atom} \times d}$ and $H_{motif}^0 \in \mathbb{R}^{N_{motif} \times d}$, respectively. N_{atom} represents the number of atoms and N_{motif} is the number of motif. The input feature of the edge in the atom-level graph between nodes p and q is $e_{pq} \in \mathbb{R}^d$.

3.3.2 Feature Extraction on Atom-Level Graph

Graph Neural Network

In the graph neural network (GNN) component, we employ a simplified message-passing mechanism that incorporates both residual connections [90] and virtual nodes [114]. At each

Table 3.3: Node features of motif-level graph.

Feature Type	Description	Feature Size
# C	Number of carbon atoms in the motif.	6
# O	Number of oxygen atoms in the motif.	6
# N	Number of nitrogen atoms in the motif.	6
# P	Number of phosphorus atoms in the motif.	6
# S	Number of sulfur atoms in the motif.	6
X	Indicates whether the motif contains a halogen atom.	1
Other Atom	Indicates whether the motif contains an atom other than H, C, O, N, P, S, or halogens.	1
# Single Bonds	Number of single bonds in the motif.	11
# Double Bonds	Number of double bonds in the motif.	8
# Triple Bonds	Number of triple bonds in the motif.	8
# Aromatic Bonds	Number of aromatic bonds in the motif.	8
Ring	Indicates whether the motif forms a ring structure.	1

GNN layer, the process starts by applying layer normalization (LN) to the node representations, followed by a $ReLU$ activation. To facilitate the exchange of global information across the graph, we introduce virtual nodes, which aggregate the features of all nodes in the graph. The resulting virtual node information is then added to the individual node representations. The operations can be formally expressed as:

$$h_p^{l+1} = ReLU(LN(h_p^l)) + vn^{l+1}, \quad (3.11)$$

$$vn^{l+1} = \sum_{k=1}^{N_{atom}} h_k^l, \quad (3.12)$$

where $h_p^{l+1} \in \mathbb{R}^d$ denotes the hidden state of node p at layer $l+1$, vn^{l+1} represents the vector of the virtual node.

Next, the message-passing step occurs, where the information from neighboring nodes and the edges connecting them is aggregated. For each edge e_{pq} , a message is computed as: $m_{pq} = h_q^{l+1} + e_{pq}$. The messages from all neighboring nodes $\mathcal{N}(p)$ are summed and used to update the

node representation through a MLP:

$$h_p^{l+1} = \text{MLP}\left(\sum_{q \in \mathcal{N}(p)} m_{pq}\right). \quad (3.13)$$

Finally, a residual connection is applied, adding the original node representation from layer l to the updated node representation at layer $l + 1$: $h_p^{l+1} \leftarrow h_p^{l+1} + h_p^l$.

Jumping Knowledge

After the GNN, we apply a jumping knowledge to aggregate information from all GNN layers. This allows each node feature to encapsulate representations from both shallow and deep layers. The operation is defined as:

$$h_p^{GNN} = \text{CONCAT}(h_p^1 A_1^T, h_p^2 A_2^T, \dots, h_p^l A_l^T), \quad (3.14)$$

where $h_p^{GNN} \in \mathbb{R}^{d_{skip} \times num_{jk}}$ represents the aggregated feature of node p from the GNN, and $A_l^T \in \mathbb{R}^{d \times d_{skip}}$ is a weight matrix that maps the layer-specific node feature $h_p^l \in \mathbb{R}^d$ to a lower-dimensional space. In our experiments, we evaluate the impact of the number of jumping knowledge layers num_{jk} on performance.

Using xLSTM to Capture the Long-Range Information

In this section, we utilize xLSTM to capture long-range dependencies for each node in the graph. We treat the output of the GNN, $H^{GNN} \in \mathbb{R}^{N_{atom} \times (d_{skip} \times num_{jk})}$, as a sequence of length N_{atom} , where each node corresponds to one element of the sequence. This sequence is then passed through the xLSTM model, producing an output $H_{atom}^{xLSTM} \in \mathbb{R}^{N_{atom} \times (d_{skip} \times num_{jk})}$, as follows:

$$H_{atom}^{xLSTM} = \text{xLSTM}(H^{GNN}). \quad (3.15)$$

3.3.3 Motif-Level Feature Extraction

The motif-level graph is processed directly by the xLSTM model. We first map the input feature H_{motif}^0 to the dimension $d_{skip} \times num_{jk}$, matching the output dimension of the atom-level graph. This mapped feature is then passed through the xLSTM model to produce an output $H_{motif}^{xLSTM} \in \mathbb{R}^{N_{motif} \times (d_{skip} \times num_{jk})}$.

$$H_{motif}^{xLSTM} = xLSTM(H_{motif}^0). \quad (3.16)$$

3.3.4 Perform a Multi-Head Mixture-of-Experts on the Features

We apply a global pooling operation to H_{GNN} , H_{atom}^{xLSTM} , and H_{motif}^{xLSTM} to produce three graph-level features: f_{GNN} , f_{atom}^{xLSTM} , and f_{motif}^{xLSTM} . The final molecular feature, f_{out} , is obtained by summing these three features. Subsequently, we utilize a Multi-Head Mixture-of-Experts (MHMoE), as described in Section 3.2.3, to process these features. For any input feature f , we split it into h_{moe} segments and the output of the MHMoE for each segment f_s is expressed as:

$$f_s^{MoE} = \sum_{i=1}^n G((f_s)_e) E_e(f_s). \quad (3.17)$$

In our work, each expert E_e is a feedforward network (FFN) consisting of a variable number of stacked fully connected layers with activation functions between them. The number of stacked layers is a hyperparameter. The final output of the MHMoE module is the concatenation of the output of all segments: $f^{MHMoE} = CONCAT(f_1^{MoE}, f_2^{MoE}, ..., f_{h_{moe}}^{MoE})$.

3.3.5 Overall architecture

The overall architecture is illustrated in Figure 3.2. We perform feature extraction on both the atom-level graph and the motif-level graph. For the atom-level graph, we first apply the GNN (Section 3.3.2), followed by a skip connection (Section 3.3.2) that aggregates the outputs from

all GNN layers, resulting in H^{GNN} . This aggregated output is then passed through the xLSTM module (Section 3.3.2), producing $H_{\text{atom}}^{\text{xLSTM}}$. Next, global pooling is applied separately to H^{GNN} and $H_{\text{atom}}^{\text{xLSTM}}$ to obtain global features of the graph from the GNN (f_{GNN}) and from the xLSTM ($f_{\text{atom}}^{\text{xLSTM}}$). These two features are then summed to generate $f_{\text{atom}} \in \mathbb{R}^{d_{\text{skip}} \times \text{num}_{jk}}$, representing the feature of the atom-level graph.

The motif-level graph is fed directly into the xLSTM model, yielding $H_{\text{motif}}^{\text{xLSTM}}$ (Section 3.3.3). We obtain a feature $f_{\text{motif}} \in \mathbb{R}^{d_{\text{skip}} \times \text{num}_{jk}}$ for the motif-level graph by applying global pooling on $H_{\text{motif}}^{\text{xLSTM}}$. And then, f_{atom} and f_{motif} are summed to form the final feature of the molecule, which is then passed through an MHMoE module (section 3.3.4) to further improve the feature representation. Finally, the final representation is passed through MLP module to predict the molecular property:

$$f_{\text{out}} = \text{MHMoE}(f_{\text{atom}} + f_{\text{motif}}), \quad (3.18)$$

$$\text{output} = \text{MLP}(f_{\text{out}}), \quad (3.19)$$

where $\text{output} \in \mathbb{R}^K$, and K represents the number of tasks.

3.3.6 Loss function

To optimize the model, we applied two losses: the task loss and the supervised contrastive loss [97]. The task loss is intended to guide the model in minimizing the error between the true label and the predicted value, while the supervised contrastive loss optimizes the feature embedding space by encouraging samples with the same label to be close to each other in the embedding space and separating those with different labels.

Task loss

For classification tasks, we use the cross-entropy loss, which measures the difference between the true label y_i and the predicted probability distribution $\hat{y}_i$. This loss is formulated as:

$$\mathcal{L}_{task}^{classification} = - \sum_{k=1}^K y_{i,k} \log(\hat{y}_{i,k}), \quad (3.20)$$

where $y_{i,k}$ represents true label and $\hat{y}_{i,k}$ is the predicted probability for task k .

For regression tasks, we adopt the mean squared error (MSE) loss, which captures the discrepancy between the predicted value $\hat{y}_i$ and the true value y_i . The MSE loss is expressed as:

$$\mathcal{L}_{task}^{regression} = (y_i - \hat{y}_i)^2. \quad (3.21)$$

Supervised contrastive loss for classification task

We apply the supervised contrastive loss (SCL) to all features: f_{out} , f_{atom} , and f_{motif} . To illustrate the process, we describe the calculation using f_{out} . First, we normalize f_{out} as follows:

$$f_{out}^{norm} = \frac{f_{out}}{\|f_{out}\|_2 + \epsilon}, \quad (3.22)$$

$$\|f_{out}\|_2 = \sqrt{\sum_d f_{out_d}^2}, \quad (3.23)$$

where ϵ is a small constant added to prevent numerical instability, and d indexes the dimensions of the feature vector.

Next, we compute the supervised contrastive loss $\mathcal{L}_{SCL}$ using the normalized feature f_{out}^{norm} :

$$\mathcal{L}_{SCL} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(f_{out_i}^{norm} \cdot f_{out_p}^{norm} / \tau)}{\sum_{a \in A(i)} \exp(f_{out_i}^{norm} \cdot f_{out_a}^{norm} / \tau)}. \quad (3.24)$$

In this equation, i denotes the index of the anchor molecule. The set $P(i)$ includes indices of all samples sharing the same label as the anchor molecule, while $A(i)$ represents the set of all sample indices excluding i . τ is a temperature parameter

Supervised contrastive loss for regression task

For the regression task, it is necessary to define positive samples. This is achieved by computing the Euclidean distance between the labels of all sample pairs in the training set. From these distances, the median value d_{med} and the maximum value d_{max} are obtained. A sample is considered as a positive sample for a given anchor if its distance to the anchor is less than d_{med} . Additionally, weights are assigned to all samples to reflect their relative importance. Samples in $P(i)$ that are closer to the anchor sample are given higher importance, while samples in $A(i)$ that are farther from the anchor sample are assigned greater importance. The loss function is formulated as follows:

$$\mathcal{L}_{SCL} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} w_p \log \frac{\exp(f_{out_i}^{norm} \cdot f_{out_p}^{norm} / \tau)}{\sum_{a \in A(i)} w_a \exp(f_{out_i}^{norm} \cdot f_{out_a}^{norm} / \tau)}, \quad (3.25)$$

where the weights are defined as:

$$w_p = \frac{(d_{med} - d_{ip})}{d_{med}}, \quad (3.26)$$

$$w_a = \exp\left(\frac{d_{ia} - d_{med}}{d_{max} - d_{med}}\right). \quad (3.27)$$

Here, d_{ip} and d_{ia} denote the Euclidean distances between sample i and sample p , and between sample i and sample a , respectively.

Overall Loss Function

The total loss function for training is a combination of the task loss and the supervised contrastive loss, given as:

$$\mathcal{L}_{total} = \mathcal{L}_{task} + \mathcal{L}_{SCL}. \quad (3.28)$$

3.4 Experiments

Table 3.4: Details of the datasets from MoleculeNet.

Dataset	# Compounds	# Tasks	Task Type
BACE	1513	1	Binary Classification
BBBP ^a	2042	1	Binary Classification
HIV	41127	1	Binary Classification
ClinTox	1478	2	Binary Classification
Sider	1427	27	Binary Classification
Tox21	7831	12	Binary Classification
Freesolv	642	1	Regression
ESOL	1128	1	Regression
Lipo	4200	1	Regression

^a 11 compounds were excluded due to parsing failures with RDKit.

Table 3.5: Details of datasets from TDC.

Dataset	# Compounds	# Tasks	Task Type
HIA	578	1	Binary Classification
Pgp	1212	1	Binary Classification
Bioavailability	640	1	Binary Classification
CYP2D6-I	13130	1	Binary Classification
CYP3A4-I	12328	1	Binary Classification
CYP2C9-I	12092	1	Binary Classification
hERG	648	1	Binary Classification
AMES	7255	1	Binary Classification
DILI	475	1	Binary Classification
Caco2	906	1	Regression
PPBR	1797	1	Regression
LD50	7385	1	Regression

3.4.1 Datasets and evaluation

MoleculeNet. MoleculeNet [36] is a widely used benchmark designed to evaluate machine learning models on molecular property prediction. We selected a subset of MoleculeNet datasets covering both classification and regression tasks.

For dataset splitting, we adopted different strategies based on task type. For single-task classification datasets, we employed scaffold splitting to ensure that molecules with different

Table 3.6: Hyperparameter setting for datasets from MoleculeNet.

	power	dimension	#experts	#heads	#expert layer
BACE	4	256	8	16	2
BBBP	4	256	8	8	2
HIV	2	128	4	8	2
ClinTox	4	128	8	8	1
Sider	4	128	8	8	1
Tox21	4	128	8	8	2
Freesolv	4	128	8	8	1
ESOL	4	256	8	8	1
Lipo	4	128	8	8	2

Table 3.7: Hyperparameter setting for datasets from TDC.

	power	dimension	#experts	#heads	#expert layer
HIA	4	32	4	4	2
Pgp	4	128	4	4	2
Bioav	4	128	4	4	2
CYP2D6-I	4	64	4	4	2
CYP3A4-I	4	128	4	4	2
CYP2C9-I	4	128	4	4	2
hERG	4	32	4	4	2
AMES	4	64	4	4	2
DILI	4	128	8	8	2
Caco2	4	128	4	4	2
PPBR	4	64	4	4	2
LD50	4	128	4	4	2

core scaffolds are separated into training, validation, and test sets. This strategy evaluates model generalization to novel chemical structures. For multi-task classification and regression datasets, we used random splitting to avoid data imbalance due to the relatively small dataset sizes.

Each dataset was split into training, validation, and test sets using an 8:1:1 ratio. The model was trained on the training set and evaluated on the validation set after each epoch. The best-performing model on the validation set was then used to report metrics on the test set. Each experiment was repeated three times, and we report the mean and standard deviation of the results.

Therapeutics Data Commons (TDC). We further evaluated our model on benchmark

datasets from the Therapeutics Data Commons (TDC) [131]. We adopted the official scaffold-based splits provided by TDC, where each dataset is partitioned into training, validation, and test sets in a 7:1:2 ratio. Each dataset includes five predefined splits. No additional resplitting or preprocessing was applied.

Evaluation Metrics. For classification tasks, we used Area Under the Receiver Operating Characteristic curve (AUROC) and Area Under the Precision-Recall Curve (AUPRC) as evaluation metrics. For regression tasks, we reported Root Mean Squared Error (RMSE) and Pearson correlation coefficient (PCC). Detailed dataset information is summarized in Table 3.4 and Table 3.5, and training hyperparameters are listed in Table 3.6 and Table 3.7.

3.4.2 Baselines

We compare our proposed method against six baseline models: Directed Message Passing Neural Network (DMPNN), Fingerprints and Graph Neural Networks (FPGNN), Hierarchical Informative Graph Neural Networks (HiGNN), Deeper Graph Convolutional Network (DeeperGCN), a transformer-based framework with focused attention (TransFoxMol), and an automated machine learning pipeline (AutoML). Each baseline represents a distinct approach to molecular representation learning or model optimization.

- **FPGNN** [111]: Combines molecular fingerprints with features derived from graph attention networks, capturing both traditional cheminformatics features and structural insights from graphs.
- **DeeperGCN** [90]: A pure graph neural network based on GCN, designed for deeper architectures to enhance feature extraction.
- **DMPNN** [61]: Optimizes message passing by centering aggregation on bonds instead of atoms, effectively encoding the chemical structure and avoiding redundant loops.
- **HiGNN** [126]: Learns molecular representations at both the atomic level and the level of substructures using hierarchical GNNs.

Table 3.8: Performance evaluation on classification datasets from MoleculeNet using AUROC and AUPRC.

	Sider		Tox21		Clintox		BBBP		BACE		HIV	
	AUC	PRC	AUC	PRC	AUC	PRC	AUC	PRC	AUC	PRC	AUC	PRC
FP-GNN	0.661 ± 0.014	0.679 ± 0.026	0.833 ± 0.004	0.459 ± 0.018	0.732 ± 0.068	0.622 ± 0.028	0.892 ± 0.019	0.953 ± 0.007	0.852 ± 0.035	0.740 ± 0.042	0.767 ± 0.039	0.328 ± 0.078
DeeperGCN	0.622 ± 0.031	0.660 ± 0.025	0.840 ± 0.010	0.434 ± 0.021	0.892 ± 0.048	0.741 ± 0.048	0.860 ± 0.014	0.937 ± 0.008	0.830 ± 0.033	0.719 ± 0.039	0.769 ± 0.041	0.300 ± 0.064
DMPNN	0.658 ± 0.032	0.680 ± 0.030	0.849 ± 0.006	0.481 ± 0.026	0.895 ± 0.010	0.727 ± 0.062	0.896 ± 0.014	0.956 ± 0.016	0.851 ± 0.028	0.742 ± 0.043	0.758 ± 0.029	0.278 ± 0.043
HiGNN	0.656 ± 0.024	0.669 ± 0.027	0.844 ± 0.006	0.462 ± 0.018	0.889 ± 0.026	0.735 ± 0.070	0.892 ± 0.014	0.943 ± 0.017	0.836 ± 0.029	0.740 ± 0.048	0.768 ± 0.038	0.310 ± 0.066
TransFoxMol	0.636 ± 0.022	0.686 ± 0.040	0.816 ± 0.011	0.367 ± 0.011	0.830 ± 0.047	0.624 ± 0.036	0.881 ± 0.015	0.947 ± 0.005	0.801 ± 0.054	0.693 ± 0.079	0.727 ± 0.037	0.232 ± 0.063
AutoML	0.682 ± 0.017	0.699 ± 0.029	0.828 ± 0.004	0.468 ± 0.022	0.875 ± 0.058	0.645 ± 0.018	0.928 ± 0.013	0.969 ± 0.019	0.840 ± 0.013	0.753 ± 0.018	0.772 ± 0.033	0.351 ± 0.078
MolGraph-xLSTM (Ours)	0.697 ± 0.022	0.713 ± 0.032	0.854 ± 0.003	0.487 ± 0.045	0.904 ± 0.032	0.714 ± 0.026	0.959 ± 0.006	0.987 ± 0.002	0.869 ± 0.016	0.784 ± 0.029	0.775 ± 0.027	0.355 ± 0.050

Table 3.9: Performance evaluation on classification datasets from MoleculeNet using Accuracy and F1 score.

	Sider		Tox21		Clintox		BBBP		BACE		HIV	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
FP-GNN	0.758 ± 0.011	0.618 ± 0.012	0.933 ± 0.003	0.384 ± 0.040	0.918 ± 0.018	0.587 ± 0.023	0.841 ± 0.036	0.893 ± 0.026	0.768 ± 0.031	0.672 ± 0.078	0.972 ± 0.001	0.354 ± 0.057
DeeperGCN	0.756 ± 0.015	0.605 ± 0.010	0.935 ± 0.001	0.290 ± 0.015	0.919 ± 0.032	0.689 ± 0.028	0.835 ± 0.018	0.892 ± 0.013	0.772 ± 0.050	0.692 ± 0.089	0.969 ± 0.002	0.330 ± 0.099
DMPNN	0.763 ± 0.016	0.625 ± 0.006	0.936 ± 0.002	0.452 ± 0.025	0.932 ± 0.007	0.624 ± 0.076	0.859 ± 0.016	0.908 ± 0.013	0.770 ± 0.031	0.689 ± 0.053	0.966 ± 0.004	0.359 ± 0.057
HiGNN	0.655 ± 0.022	0.612 ± 0.035	0.860 ± 0.011	0.392 ± 0.017	0.866 ± 0.029	0.665 ± 0.032	0.825 ± 0.010	0.875 ± 0.010	0.781 ± 0.038	0.703 ± 0.058	0.849 ± 0.040	0.197 ± 0.052
TranFoxMol	0.756 ± 0.019	0.651 ± 0.032	0.939 ± 0.003	0.324 ± 0.042	0.940 ± 0.004	0.656 ± 0.024	0.856 ± 0.004	0.908 ± 0.005	0.711 ± 0.095	0.488 ± 0.191	0.973 ± 0.001	0.153 ± 0.057
AutoML	0.770 ± 0.008	0.637 ± 0.017	0.938 ± 0.002	0.385 ± 0.004	0.937 ± 0.004	0.549 ± 0.108	0.895 ± 0.017	0.934 ± 0.010	0.768 ± 0.039	0.664 ± 0.074	0.973 ± 0.001	0.331 ± 0.063
MolGraph-xLSTM (Ours)	0.762 ± 0.012	0.647 ± 0.014	0.937 ± 0.002	0.421 ± 0.070	0.934 ± 0.002	0.647 ± 0.026	0.896 ± 0.020	0.931 ± 0.014	0.789 ± 0.033	0.692 ± 0.063	0.973 ± 0.001	0.357 ± 0.083

- **TransFoxMol** [112]: Integrates the power of graph neural networks and transformers to capture global and local molecular features efficiently.
- **AutoML**: A model selection and optimization pipeline based on automated machine learning techniques. It ensembles multiple algorithms and performs hyperparameter tuning automatically. In our experiments, we used H2O AutoML [132], which includes tree-based models such as XGBoost, GBM, and stacked ensembles.

Table 3.10: Performance evaluation on regression datasets from MoleculeNet using RMSE and Pearson Correlation (PCC).

	ESOL		Lipo		Freesolv	
	RMSE	PCC	RMSE	PCC	RMSE	PCC
FP-GNN	0.658 ±0.006	0.946 ±0.006	0.610 ±0.028	0.861 ±0.012	1.106 ±0.195	0.951 ±0.023
DeeperGCN	0.615 ±0.044	0.954 ±0.008	0.645 ±0.048	0.842 ±0.026	1.261 ±0.022	0.938 ±0.007
DMPNN	0.575 ±0.073	0.957 ±0.015	0.553 ±0.033	0.842 ±0.026	1.211 ±0.120	0.945 ±0.007
HiGNN	0.570 ±0.061	0.959 ±0.013	0.563 ±0.041	0.882 ±0.018	1.068 ±0.092	0.956 ±0.007
TransFoxMol	0.930 ±0.261	0.917 ±0.047	0.652 ±0.033	0.855 ±0.011	1.225 ±0.155	0.945 ±0.007
AutoML	0.843 ±0.062	0.910 ±0.023	0.792 ±0.043	0.748 ±0.031	1.235 ±0.220	0.941 ±0.024
MolGraph- xLSTM (Ours)	0.527 ±0.046	0.965 ±0.010	0.550 ±0.026	0.888 ±0.011	1.024 ±0.076	0.960 ±0.006

Table 3.11: Performance evaluation on regression datasets from MoleculeNet using MAE and coefficient of determination (R2).

	ESOL		Lipo		Freesolv	
	MAE	R2	MAE	R2	MAE	R2
FP-GNN	0.661 ±0.014	0.893 ±0.012	0.457 ±0.017	0.738 ±0.020	0.678 ±0.159	0.902 ±0.046
DeeperGCN	0.470 ±0.025	0.906 ±0.018	0.479 ±0.028	0.705 ±0.044	0.692 ±0.096	0.878 ±0.013
DMPNN	0.416 ±0.040	0.916 ±0.029	0.404 ±0.029	0.784 ±0.030	0.729 ±0.056	0.888 ±0.018
HiGNN	0.408 ±0.041	0.918 ±0.024	0.423 ±0.023	0.776 ±0.031	0.578 ±0.100	0.912 ±0.016
TranFoxMol	0.709 ±0.192	0.780 ±0.101	0.500 ±0.042	0.645 ±0.088	0.800 ±0.115	0.877 ±0.019
AutoML	0.625 ±0.042	0.823 ±0.035	0.605 ±0.030	0.559 ±0.048	0.758 ±0.109	0.879 ±0.048
MolGraph- xLSTM (Ours)	0.385 ±0.030	0.930 ±0.019	0.412 ±0.022	0.787 ±0.019	0.607 ±0.071	0.920 ±0.014

3.4.3 Experimental results

MoleculeNet

MolGraph-xLSTM demonstrates improved performance across both classification and regression datasets, highlighting its robustness in handling diverse molecular property prediction tasks. In the classification tasks (Table 3.8 and Table 3.9), MolGraph-xLSTM achieves particularly strong results on the Sider and BBBP datasets. For the Sider dataset, MolGraph-xLSTM achieves an AUROC of 0.697 ± 0.022 , representing a 5.45% improvement over the best baseline, FP-GNN (0.661 ± 0.014). Similarly, on the BBBP dataset, MolGraph-xLSTM achieves an AUROC of 0.959 ± 0.006 , which is a 7.03% improvement compared to the best baseline, TransFoxMol (0.896 ± 0.024).

For regression datasets (Table 3.10 and Table 3.11), MolGraph-xLSTM delivers competitive performance across multiple benchmarks. On the ESOL dataset, MolGraph-xLSTM achieves an RMSE of 0.527 ± 0.046 , reflecting a 7.54% improvement over the best-performing baseline, HiGNN (0.570 ± 0.061). On the FreeSolv dataset, MolGraph-xLSTM achieves the lowest RMSE of 1.024 ± 0.076 and the highest PCC of 0.960 ± 0.006 , demonstrating its reliability in regression tasks.

TDC Benchmark

MolGraph-xLSTM exhibits consistent performance across both classification and regression tasks in the TDC benchmark, indicating its capacity to generalize across diverse pharmacological endpoints. In classification tasks (Tables 3.12–3.15), MolGraph-xLSTM achieves the highest average AUROC (0.866) and AUPRC (0.861) across nine classification datasets, slightly outperforming competitive baselines such as DMPNN (AUROC: 0.861, AUPRC: 0.853) and FPGNN (AUROC: 0.859, AUPRC: 0.856).

MolGraph-xLSTM achieves noticeable improvement on the Bioavailability dataset, which measures the fraction of an administered drug that reaches systemic circulation. It obtains

Table 3.12: Performance evaluation on classification datasets from TDC using AUROC.

	HIA	Pgp	Bioavail.	CYP2D6-I	CYP3A4-I	CYP2C9-I	hERG	AMES	DILI
FPGNN	0.958 ± 0.012	0.930 ± 0.007	0.666 ± 0.035	0.863 ± 0.005	0.869 ± 0.004	0.867 ± 0.002	0.846 ± 0.029	0.836 ± 0.006	0.899 ± 0.017
DeeperGCN	0.965 ± 0.017	0.884 ± 0.009	0.579 ± 0.113	0.850 ± 0.005	0.883 ± 0.004	0.871 ± 0.002	0.734 ± 0.050	0.818 ± 0.010	0.864 ± 0.026
DMPNN	0.976 ± 0.004	0.889 ± 0.005	0.617 ± 0.050	0.872 ± 0.004	0.887 ± 0.003	0.878 ± 0.004	0.741 ± 0.024	0.838 ± 0.012	0.883 ± 0.021
HiGNN	0.974 ± 0.007	0.882 ± 0.022	0.620 ± 0.070	0.867 ± 0.007	0.886 ± 0.006	0.878 ± 0.009	0.807 ± 0.027	0.822 ± 0.014	0.892 ± 0.011
TransFoxMol	0.951 ± 0.036	0.875 ± 0.011	0.619 ± 0.019	0.859 ± 0.004	0.854 ± 0.009	0.867 ± 0.004	0.799 ± 0.017	0.816 ± 0.011	0.896 ± 0.024
AutoML	0.880 ± 0.012	0.821 ± 0.007	0.509 ± 0.026	0.762 ± 0.011	0.773 ± 0.004	0.776 ± 0.010	0.682 ± 0.021	0.756 ± 0.008	0.766 ± 0.017
MolGraph-xLSTM (Ours)	0.977 ± 0.010	0.908 ± 0.011	0.684 ± 0.118	0.872 ± 0.002	0.890 ± 0.005	0.881 ± 0.006	0.846 ± 0.017	0.832 ± 0.011	0.904 ± 0.006

Table 3.13: Performance evaluation on classification datasets from TDC using AUPRC.

	HIA	Pgp	Bioavail.	CYP2D6-I	CYP3A4-I	CYP2C9-I	hERG	AMES	DILI
FPGNN	0.985 ± 0.006	0.936 ± 0.008	0.875 ± 0.012	0.641 ± 0.020	0.840 ± 0.007	0.720 ± 0.005	0.929 ± 0.021	0.878 ± 0.009	0.896 ± 0.025
DeeperGCN	0.990 ± 0.005	0.900 ± 0.011	0.811 ± 0.070	0.613 ± 0.009	0.852 ± 0.005	0.741 ± 0.010	0.869 ± 0.037	0.861 ± 0.010	0.865 ± 0.018
DMPNN	0.993 ± 0.001	0.904 ± 0.008	0.849 ± 0.022	0.644 ± 0.010	0.858 ± 0.004	0.750 ± 0.006	0.857 ± 0.025	0.875 ± 0.013	0.872 ± 0.032
HiGNN	0.991 ± 0.003	0.885 ± 0.022	0.850 ± 0.042	0.633 ± 0.013	0.853 ± 0.009	0.741 ± 0.020	0.918 ± 0.010	0.856 ± 0.018	0.870 ± 0.028
TransFoxMol	0.983 ± 0.016	0.890 ± 0.013	0.840 ± 0.023	0.603 ± 0.014	0.827 ± 0.011	0.721 ± 0.013	0.907 ± 0.010	0.860 ± 0.009	0.896 ± 0.013
AutoML	0.933 ± 0.006	0.769 ± 0.011	0.761 ± 0.010	0.444 ± 0.015	0.652 ± 0.007	0.558 ± 0.002	0.814 ± 0.010	0.760 ± 0.007	0.710 ± 0.019
MolGraph-xLSTM (Ours)	0.992 ± 0.003	0.919 ± 0.009	0.872 ± 0.057	0.659 ± 0.007	0.863 ± 0.007	0.757 ± 0.006	0.933 ± 0.006	0.870 ± 0.010	0.888 ± 0.006

Table 3.14: Performance evaluation on classification datasets from TDC using Accuracy.

	HIA	Pgp	Bioav	CYP2D6-I	CYP3A4-I	CYP2C9-I	hERG	AMES	DILI
HiGNN	0.945 ± 0.010	0.791 ± 0.013	0.658 ± 0.081	0.825 ± 0.006	0.798 ± 0.010	0.806 ± 0.020	0.776 ± 0.042	0.762 ± 0.006	0.827 ± 0.016
FPGNN	0.908 ± 0.028	0.850 ± 0.017	0.713 ± 0.047	0.870 ± 0.015	0.773 ± 0.016	0.806 ± 0.005	0.791 ± 0.034	0.770 ± 0.007	0.810 ± 0.028
DMPNN	0.920 ± 0.037	0.815 ± 0.019	0.673 ± 0.044	0.876 ± 0.007	0.794 ± 0.007	0.815 ± 0.006	0.797 ± 0.035	0.760 ± 0.019	0.821 ± 0.019
TransFoxMol	0.868 ± 0.048	0.788 ± 0.014	0.616 ± 0.074	0.879 ± 0.003	0.753 ± 0.017	0.794 ± 0.006	0.689 ± 0.211	0.742 ± 0.029	0.664 ± 0.181
DeeperGCN	0.918 ± 0.016	0.805 ± 0.013	0.719 ± 0.044	0.868 ± 0.004	0.788 ± 0.009	0.807 ± 0.008	0.783 ± 0.028	0.760 ± 0.010	0.802 ± 0.024
AutoML	0.944 ± 0.008	0.821 ± 0.007	0.698 ± 0.031	0.873 ± 0.004	0.766 ± 0.006	0.784 ± 0.004	0.802 ± 0.019	0.773 ± 0.005	0.771 ± 0.016
MolGraph-xLSTM (Ours)	0.950 ± 0.022	0.806 ± 0.019	0.755 ± 0.012	0.876 ± 0.003	0.800 ± 0.006	0.811 ± 0.010	0.832 ± 0.016	0.763 ± 0.011	0.817 ± 0.056

Table 3.15: Performance evaluation on classification datasets from TDC using F1 score.

	HIA	Pgp	Bioav	CYP2D6-I	CYP3A4-I	CYP2C9-I	hERG	AMES	DILI
HiGNN	0.965 ± 0.006	0.781 ± 0.012	0.761 ± 0.068	0.591 ± 0.006	0.784 ± 0.005	0.724 ± 0.013	0.849 ± 0.031	0.798 ± 0.006	0.845 ± 0.014
FPGNN	0.943 ± 0.016	0.852 ± 0.019	0.819 ± 0.046	0.583 ± 0.027	0.751 ± 0.028	0.696 ± 0.013	0.871 ± 0.014	0.812 ± 0.013	0.827 ± 0.033
DMPNN	0.949 ± 0.022	0.818 ± 0.026	0.793 ± 0.037	0.626 ± 0.012	0.775 ± 0.015	0.707 ± 0.027	0.871 ± 0.026	0.795 ± 0.029	0.836 ± 0.017
TransFoxMol	0.912 ± 0.033	0.782 ± 0.023	0.719 ± 0.085	0.559 ± 0.026	0.689 ± 0.046	0.652 ± 0.025	0.710 ± 0.331	0.789 ± 0.037	0.489 ± 0.447
DeeperGCN	0.946 ± 0.011	0.804 ± 0.021	0.824 ± 0.035	0.574 ± 0.017	0.779 ± 0.004	0.695 ± 0.016	0.862 ± 0.022	0.805 ± 0.006	0.805 ± 0.032
AutoML	0.965 ± 0.005	0.822 ± 0.006	0.815 ± 0.023	0.613 ± 0.016	0.757 ± 0.002	0.690 ± 0.010	0.874 ± 0.014	0.816 ± 0.003	0.798 ± 0.014
MolGraph-xLSTM (Ours)	0.968 ± 0.014	0.795 ± 0.035	0.851 ± 0.017	0.591 ± 0.054	0.776 ± 0.005	0.704 ± 0.024	0.893 ± 0.010	0.809 ± 0.007	0.840 ± 0.034

an AUROC of 0.684 ± 0.118 , compared to 0.666 ± 0.035 from the best-performing baseline (FPGNN), and maintains a competitive AUPRC of 0.872 ± 0.057 .

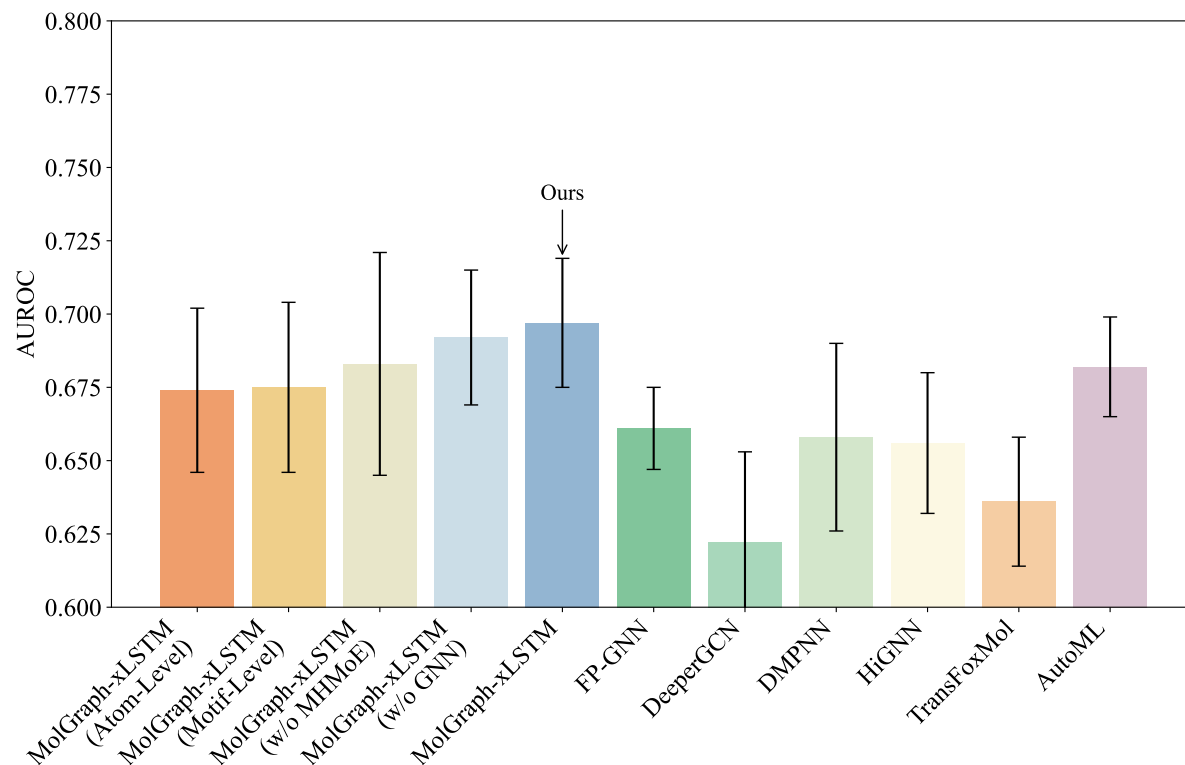
In regression tasks (Table 3.16 and Table 3.17), MolGraph-xLSTM achieves leading or comparable results. It obtains the lowest RMSE on both the Caco2 (0.358 ± 0.015) and PPBR (11.772 ± 0.200) datasets, reflecting 11.17% and 3.81% improvements over the next-best models. Additionally, it achieves the highest PCC of 0.861 ± 0.011 on Caco2 and 0.644 ± 0.019 on PPBR.

Table 3.16: Performance evaluation on regression datasets from TDC using RMSE and Pearson Correlation (PCC).

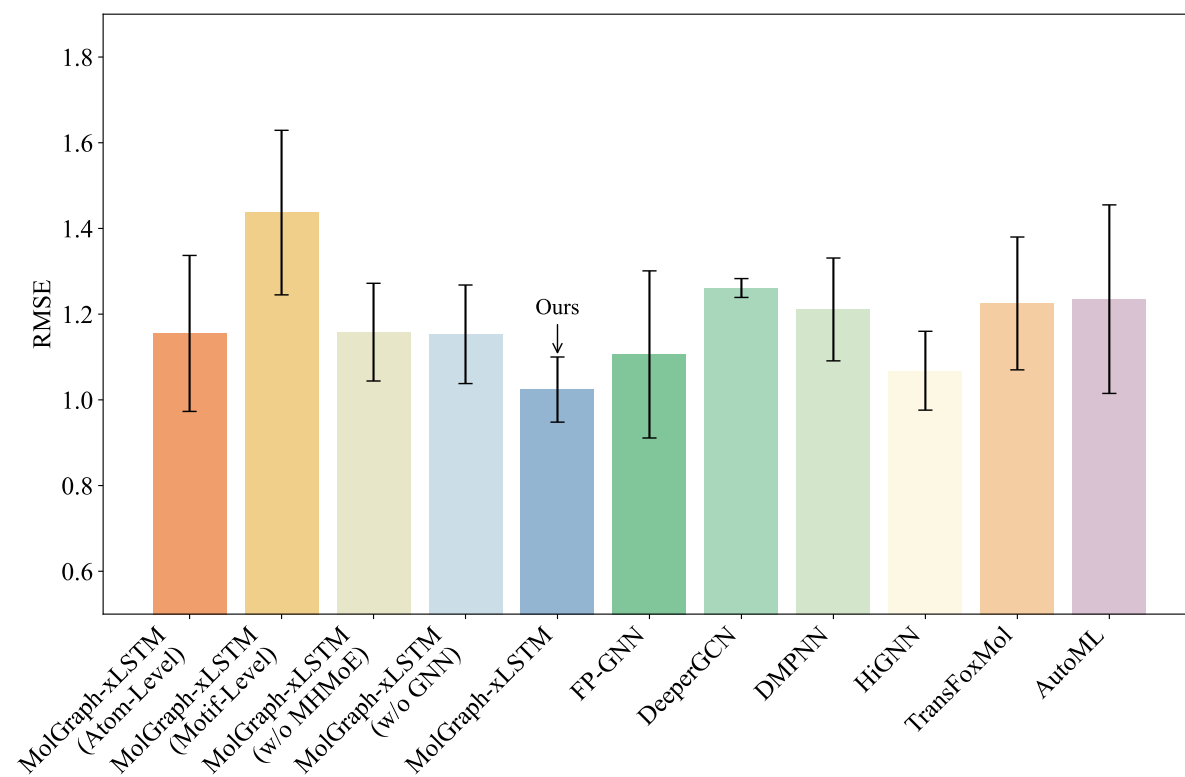
	Caco2		PPBR		LD50	
	RMSE	PCC	RMSE	PCC	RMSE	PCC
FP-GNN	0.408 ± 0.056	0.835 ± 0.027	12.238 ± 1.194	0.614 ± 0.068	0.911 ± 0.040	0.544 ± 0.046
DeeperGCN	0.624 ± 0.034	0.470 ± 0.112	14.634 ± 0.392	0.305 ± 0.040	0.951 ± 0.063	0.472 ± 0.109
DMPNN	0.487 ± 0.103	0.796 ± 0.023	12.497 ± 0.230	0.588 ± 0.031	0.859 ± 0.035	0.608 ± 0.033
HiGNN	0.457 ± 0.064	0.794 ± 0.043	13.247 ± 0.724	0.554 ± 0.056	0.941 ± 0.038	0.523 ± 0.034
TransFoxMol	0.596 ± 0.082	0.719 ± 0.071	13.638 ± 0.349	0.512 ± 0.027	0.922 ± 0.053	0.538 ± 0.066
AutoML	0.403 ± 0.014	0.820 ± 0.009	13.565 ± 0.139	0.471 ± 0.016	0.841 ± 0.011	0.622 ± 0.012
MolGraph-xLSTM (Ours)	0.358 ± 0.015	0.861 ± 0.011	11.772 ± 0.200	0.644 ± 0.019	0.871 ± 0.026	0.600 ± 0.026

Table 3.17: Performance evaluation on regression datasets from TDC using MAE and R2.

	Caco2		PPBR		LD50	
	MAE	R2	MAE	R2	MAE	R2
FP-GNN	0.326 ± 0.040	0.642 ± 0.100	8.465 ± 1.709	0.343 ± 0.135	0.638 ± 0.024	0.258 ± 0.065
DeeperGCN	0.513 ± 0.016	0.170 ± 0.088	10.911 ± 0.686	0.067 ± 0.050	0.673 ± 0.053	0.190 ± 0.107
DMPNN	0.388 ± 0.077	0.478 ± 0.231	8.158 ± 0.314	0.320 ± 0.025	0.607 ± 0.022	0.340 ± 0.054
HiGNN	0.364 ± 0.048	0.550 ± 0.135	9.747 ± 0.786	0.234 ± 0.085	0.657 ± 0.018	0.209 ± 0.063
TranFoxMol	0.487 ± 0.068	0.233 ± 0.220	9.055 ± 0.523	0.189 ± 0.041	0.645 ± 0.036	0.239 ± 0.088
AutoML	0.327 ± 0.012	0.656 ± 0.023	9.533 ± 0.162	0.198 ± 0.016	0.589 ± 0.006	0.368 ± 0.017
MolGraph-xLSTM (Ours)	0.286 ± 0.009	0.727 ± 0.022	7.672 ± 0.207	0.396 ± 0.021	0.614 ± 0.013	0.323 ± 0.040



(a) Sider dataset



(b) FreeSolv dataset

Figure 3.3: **Ablation study results on the Sider and FreeSolv datasets.** Performance comparison of ablation variants against the full MolGraph-xLSTM model and baseline models.

Table 3.18: Ablation results for MolGraph-xLSTM on the Sider (classification) and FreeSolv (regression) datasets.

	Sider		Freesolv	
	AUROC	AUPRC	RMSE	PCC
MolGraph-xLSTM (Atom-Level)	0.674 ± 0.028	0.697 ± 0.034	1.155 ± 0.182	0.947 ± 0.017
MolGraph-xLSTM (Motif-Level)	0.675 ± 0.029	0.699 ± 0.030	1.437 ± 0.192	0.924 ± 0.015
MolGraph-xLSTM (w/o MHMoE)	0.683 ± 0.038	0.704 ± 0.034	1.158 ± 0.114	0.949 ± 0.010
MolGraph-xLSTM (w/o GNN)	0.692 ± 0.023	0.708 ± 0.031	1.153 ± 0.115	0.949 ± 0.014

3.4.4 Ablation Study

Effect of Different Designed Modules

We conducted an ablation study to evaluate the contributions of different components in MolGraph-xLSTM, including the atom-level branch (MolGraph-xLSTM (Atom-Level)), motif-level branch (MolGraph-xLSTM (Motif-Level)), multi-head mixture-of-experts module (MolGraph-xLSTM(w/o MHMoE)), and the GNN component within the atom-level branch (MolGraph-xLSTM(w/o GNN)). The results, presented in Table 3.18 and Figure 3.3, highlight the importance of these components in achieving superior performance.

The full MolGraph-xLSTM model consistently outperformed all ablation variants, highlighting the effectiveness of its integrated architecture. Notably, even with only the atom-level branch, MolGraph-xLSTM achieved competitive performance, outperforming other atom-level graph-based models like DMPNN and DeeperGCN, as well as TransFoxMol, a hybrid model integrating GNN and Transformer. These results validate the design of our hybrid GNN and xLSTM framework as an effective approach for molecular representation learning. For the motif-level branch, it also outperformed other baselines on the Sider dataset, including HiGNN, which also utilizes motif-level graphs, in the classification task. However, its performance on the regression dataset was suboptimal. This suggests that the motif-level initialization features utilized in our model may not sufficiently capture the granularity required for regression tasks,

highlighting opportunities for further improvement.

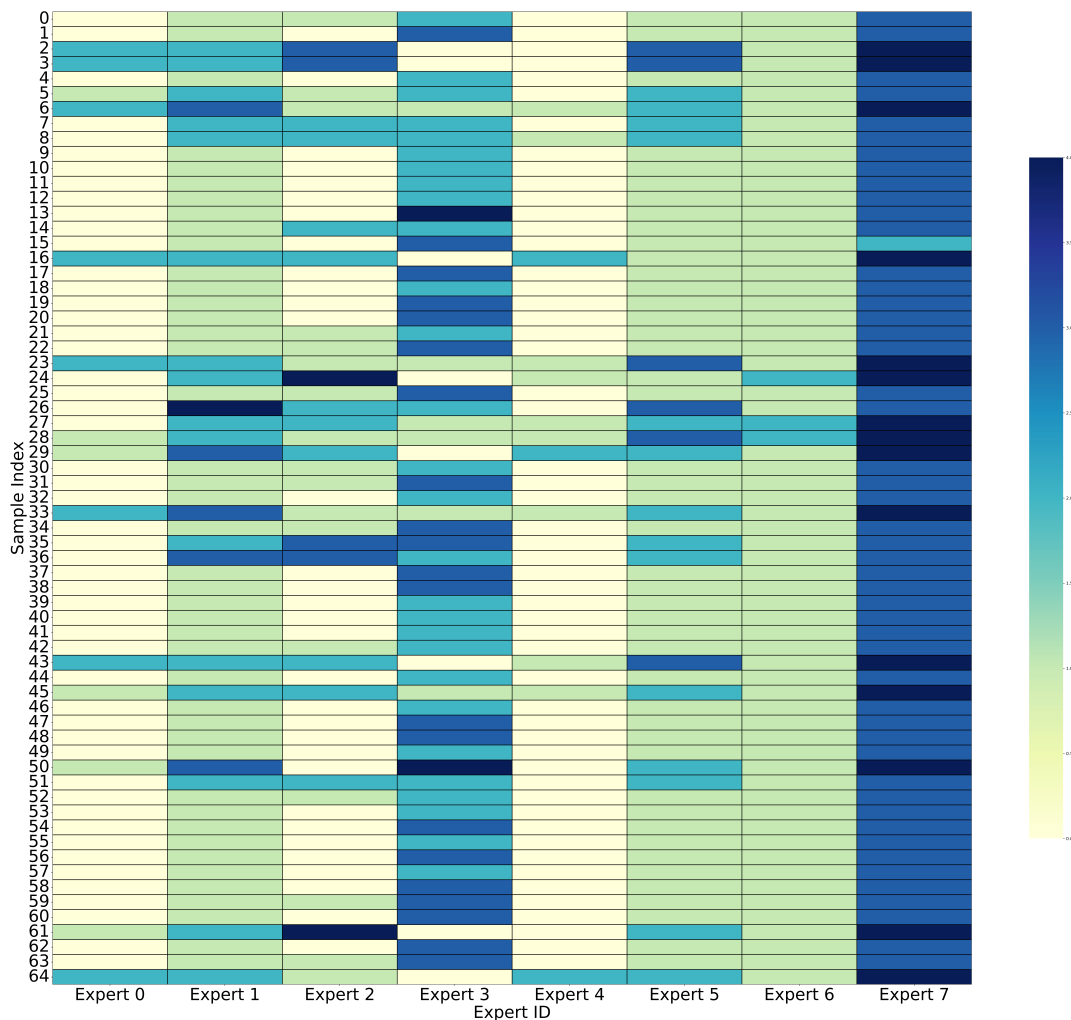


Figure 3.4: Activation patterns of samples across the experts in the MHMoE module on the Freesolv dataset. Each row represents a sample, and each column corresponds to an expert.

The MHMoE module contributed to the model performance, particularly on the FreeSolv dataset. Removing the MHMoE module resulted in an RMSE increase from 1.024 to 1.158, closely aligning with the performance of the atom-level-only variant, indicating its role in improving regression performance. As shown in Figure 3.4 and Figure 3.5, the activation maps demonstrate that all experts actively contribute to the task, indicating effective load balancing. This balanced activation ensures no single expert is overwhelmed, allowing the network to fully leverage the diverse expertise of all experts.

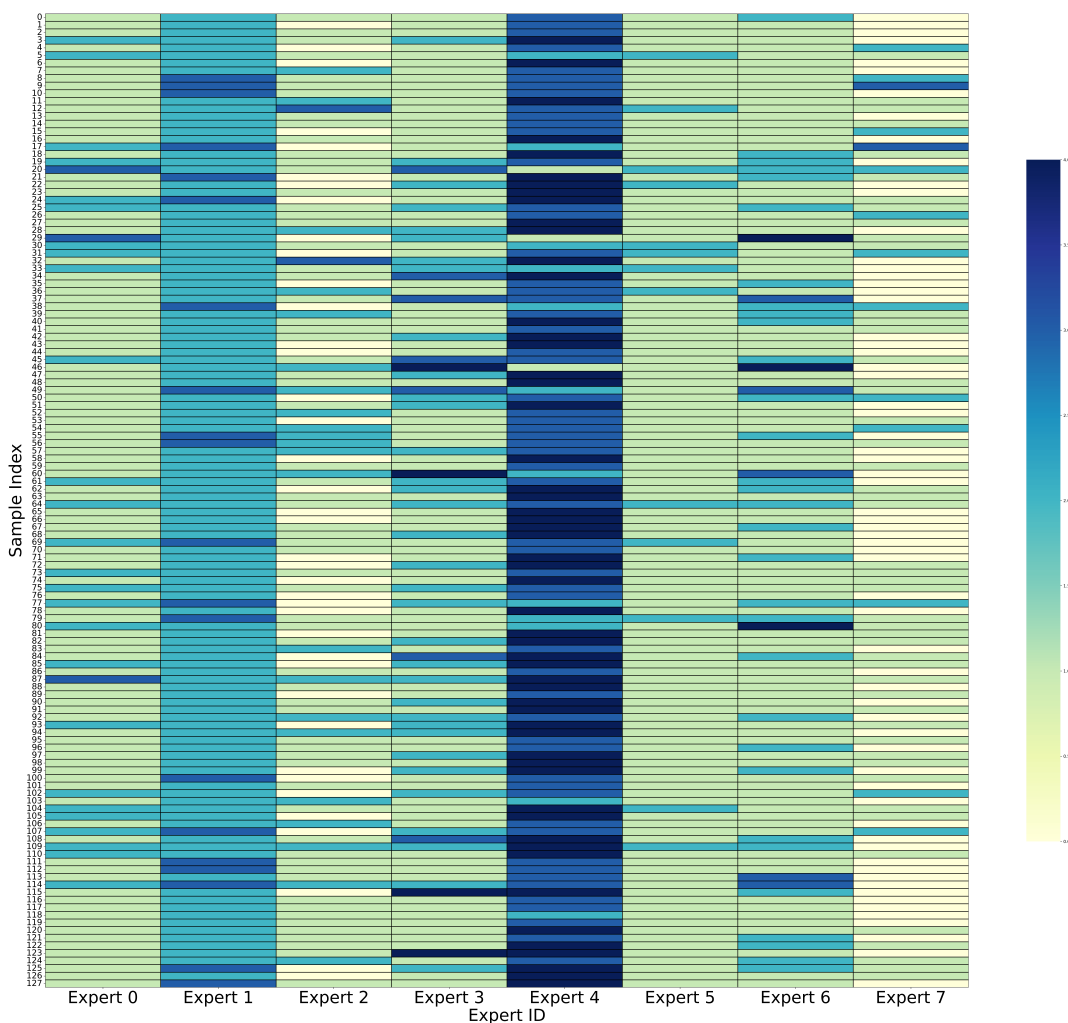


Figure 3.5: **Activation patterns of samples across the experts in the MHMoE module on the Sider dataset.** Each row represents a sample, and each column corresponds to an expert.

Among the four components, the GNN had the least impact on the Sider dataset but showed a notable influence on FreeSolv. Overall, the ablation study demonstrates that the atom- and motif-level branches provide complementary insights into molecular representation learning, and their integration enhances the model performance. This highlights the effectiveness of the proposed approach for molecular modeling.

Impact of Node Input Order for Molecular Graphs on Performance

xLSTM is originally designed for sequence data, which inherently has a fixed order. However, graph data does not have this property, as it can start from any node (Figure 3.6). In our initial

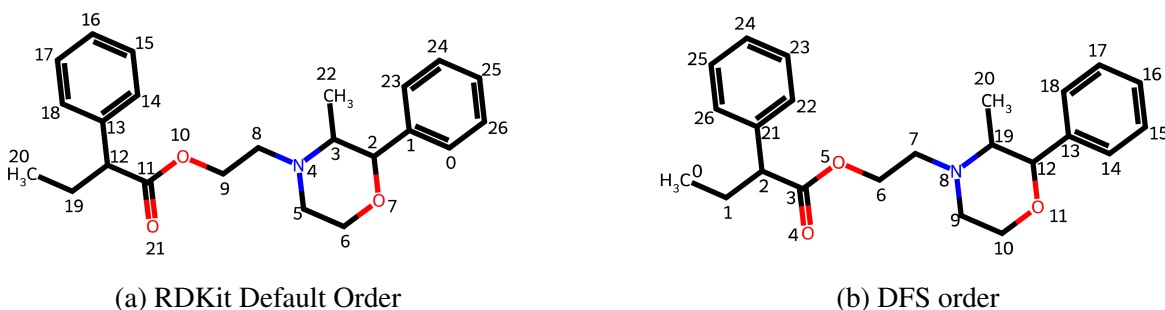


Figure 3.6: **Examples of different atom input orders for molecular graphs in xLSTM.** (a) RDKit Default Order: Atoms are ordered as per the default output from RDKit. (b) DFS Order: Atoms are ordered based on a Depth-First Search traversal of the molecular graph.

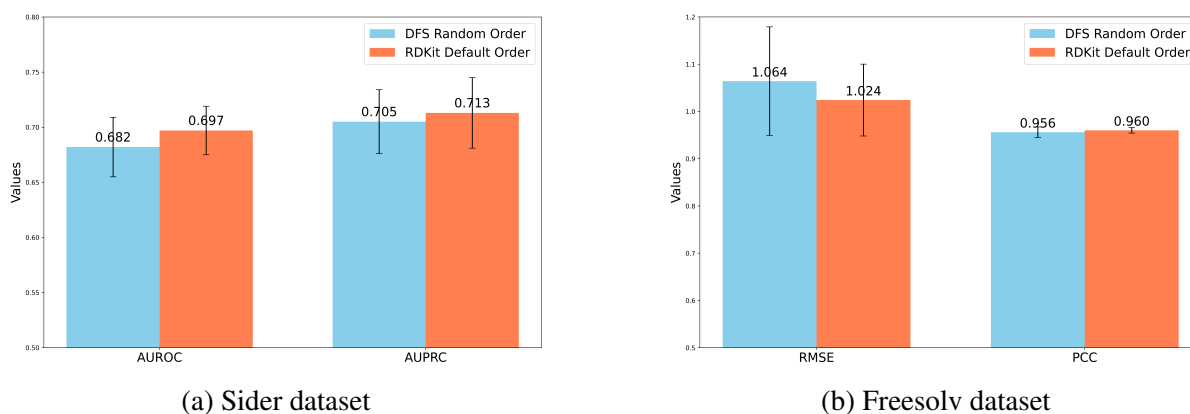


Figure 3.7: **Performance comparison of MolGraph-xLSTM trained with different node orderings.** The RDKit Default Order refers to the node sequence provided by RDKit, while the DFS Random Order generates a sequence by performing a depth-first search starting from a randomly selected node. (a) Results on the Sider dataset (classification) using AUROC and AUPRC as metrics. (b) Results on the FreeSolv dataset (regression) using RMSE and PCC as metrics.

tests, we used the default node order provided by RDKit. In this section, we evaluate the effect of using a randomized starting node during training. Specifically, we generate the node sequence by performing a depth-first search (DFS) starting from a randomly selected initial node in the graph for each training instance.

Figure 3.7 compares the performance of MolGraph-xLSTM trained with the RDKit default node order and the DFS random order on Sider and Freesolv datasets. On the Sider dataset (Figure 3.7 (a)), the model trained with the RDKit default order slightly outperformed the DFS random order in both AUROC and AUPRC metrics. Similarly, on the FreeSolv dataset (Figure 3.7 (b)), the RMSE and PCC metrics indicate a marginal advantage for the RDKit default order. Despite these differences, the results show that MolGraph-xLSTM achieves competitive performance with both node orderings. This suggests that the model is robust to changes in the input node sequence.

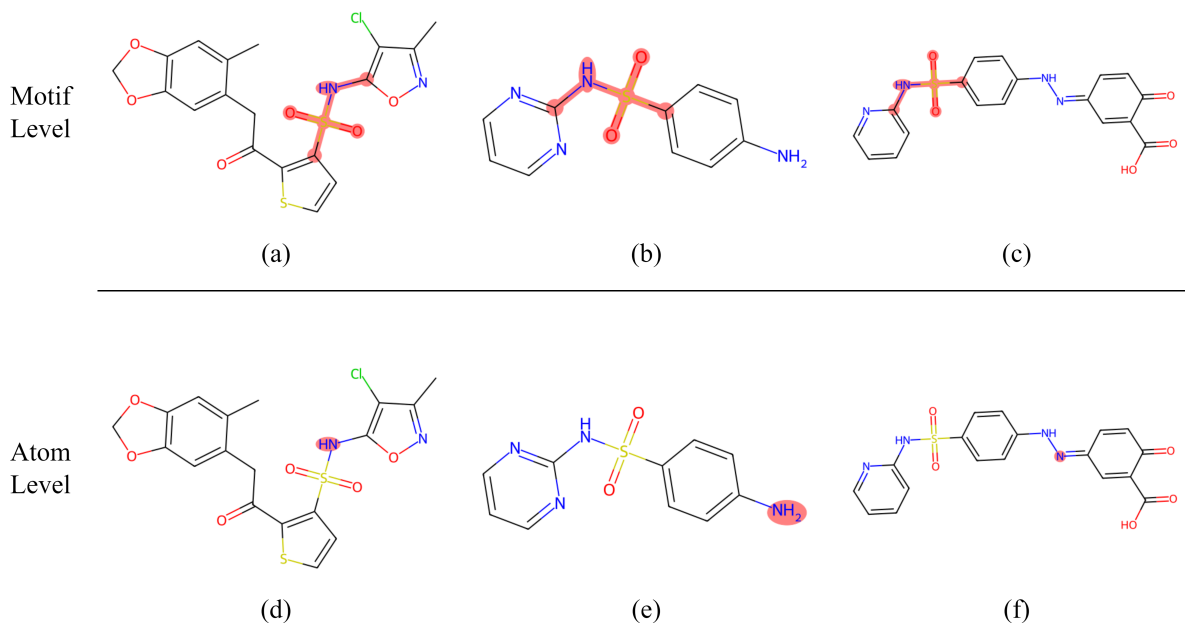


Figure 3.8: **Visualization of the highest-weighted motifs and atoms identified by the model for molecules from the Sider test set containing the SO_2NH substructure.** The top row highlights the motifs with the highest attention weights from the motif-level branch, while the bottom row highlights the atoms with the highest attention weights from the atom-level branch.

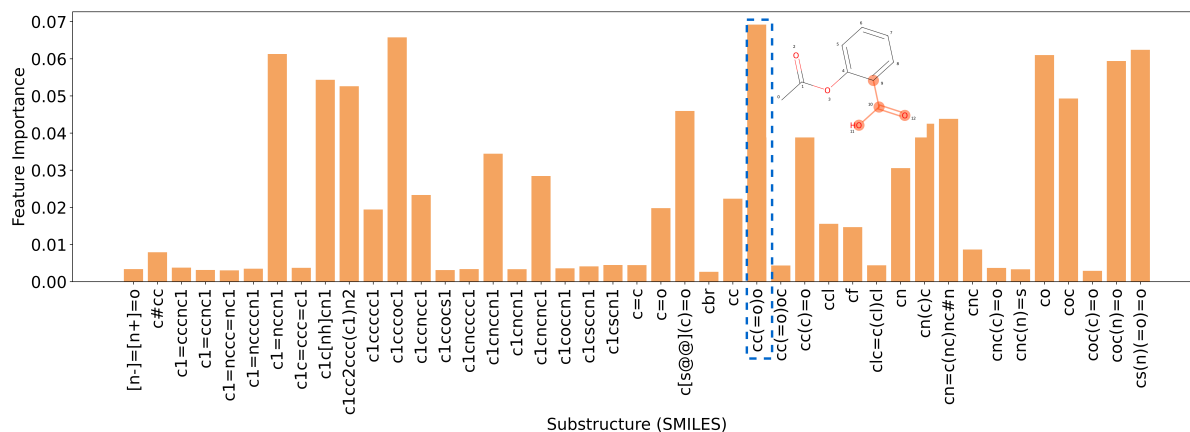


Figure 3.9: **Importance scores of substructures identified by MolGraph-xLSTM on the BBBP dataset.** For each molecule, the substructure with the highest model-assigned weight was analyzed using a random forest model to determine its relationship with BBBP labels. The substructure $-CC(=O)O-$, containing a carboxylic group, received the highest importance score (highlighted by the blue dashed box).

3.4.5 Interpretability analysis

To evaluate the interpretability of MolGraph-xLSTM, we visualized the motifs and atomic sites with the highest model-assigned weights from the motif-level and atom-level networks. By applying max-pooling to the output of the xLSTM layer, we identified the features with the greatest contributions, providing us insight into the substructures and atomic sites that are most closely related to the properties of a particular molecule.

In Figure 3.8, all three molecules highlight the $-SO_2NH-$ (sulfonamide) substructure, a chemical motif known to be strongly linked with adverse reactions such as Type IV hypersensitivity, blurred vision, and other side effects [133]. These adverse effects correspond to side effects labeled in the Sider dataset, including Eye Disorders, Immune System Disorders, and Skin and Subcutaneous Tissue Disorders, demonstrating an alignment between the highlighted substructure and known biological properties of sulfonamides. Additionally, molecules like Figure 3.8(e) and Figure 3.8(f) emphasize atomic sites beyond the sulfonamide motif. In Figure 3.8(f), the highlighted N atom resides within the hydrazine group ($-NH-N=$), which is known to exert toxic effects on multiple organ systems, including neurological, hematological and pulmonary [134]. This suggests that the atom-level network captures additional

fine-grained features that complement the broader motif-level representations, demonstrating the capacity of the model to integrate complementary information from both atom-level and motif-level networks.

We further conducted an analysis on the BBBP dataset (Blood-Brain Barrier Permeability), a crucial property in evaluating the ability of a drug to cross the blood-brain barrier and target central nervous system (CNS) disorders. Accurate prediction of this property is essential for developing CNS-targeted therapies. For each molecule in the dataset, the substructure with the highest weight assigned by MolGraph-xLSTM was identified. These substructures were further analyzed using a random forest model [135] to determine their relationship with BBBP labels.

Figure 3.9 illustrates the importance scores of substructures as determined by the random forest model. Among these, the substructure $-CC(=O)O-$, containing a carboxylic group ($-C(=O)O-$), achieved the highest importance score. This finding is supported by previous studies [136, 137], which have highlighted the role of the carboxylic group in influencing BBBP.

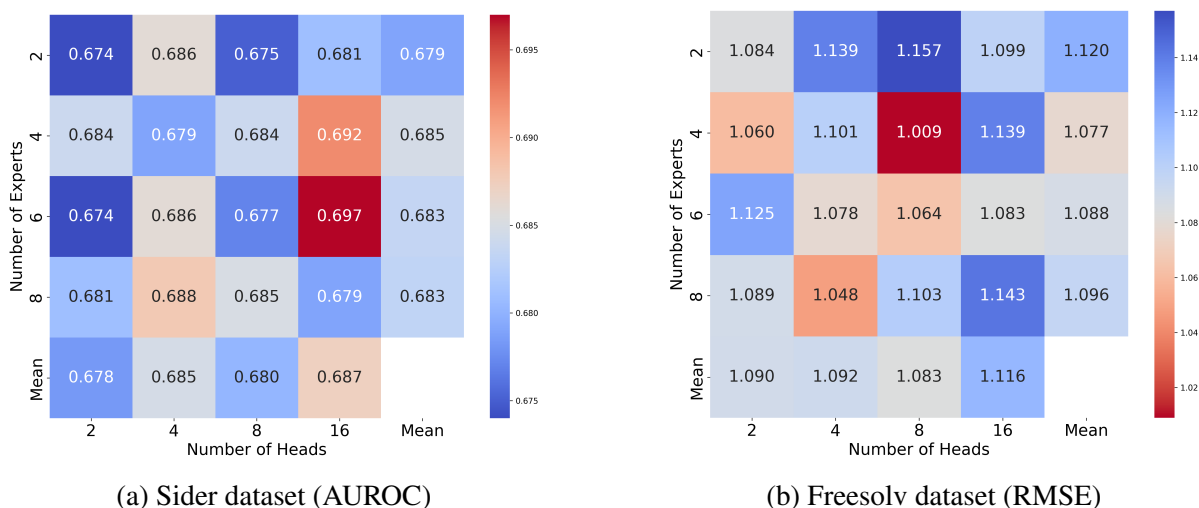


Figure 3.10: **Analysis of the impact of the number of experts and heads on the Sider and FreeSolv datasets.** (a) Performance on the Sider dataset measured by AUROC, where red indicates higher AUROC (better performance) and blue represents lower AUROC (worse performance). (b) Performance on the FreeSolv dataset measured by RMSE, with red corresponding to lower RMSE (better performance) and blue indicating higher RMSE (worse performance).

3.4.6 Hyperparameter analysis

Performance of MolGraph-xLSTM with Varing Number of Experts and Heads in the MHMoE

The heatmaps in Figure 3.10 reveal the impact of the number of experts and heads in the MHMoE module on the model's performance for the Sider and FreeSolv datasets. For both datasets, configurations with 2 experts generally perform poorly, while increasing the number of experts to 4 or 6 yields better results. Beyond 6 experts, no significant improvements are observed, suggesting that additional experts may become redundant for these datasets, as they do not process substantially different information.

For the Sider dataset, measured by AUROC, an increase in the number of heads consistently enhances performance, indicating that more heads improve the model's ability to handle classification tasks. In contrast, for the FreeSolv dataset, measured by RMSE, increasing the number of heads beyond 8 leads to a noticeable decline in performance, particularly when the number of heads reaches 16. This decline is likely due to overfitting, as FreeSolv is a relatively small dataset. These observations highlight the need to balance the number of experts and heads based on the task and dataset size, as excessive complexity can negatively affect performance.

Performance of MolGraph-xLSTM with Varing Number of Jump Layers

The results in Figure 3.11 illustrate the impact of varying the number of jump layers on the performance of MolGraph-xLSTM across the Sider and FreeSolv datasets. On the Sider dataset, the AUROC shows relatively small fluctuations, with the maximum value of 0.697 observed at 4 jump layers and the minimum value of 0.673 at 8 jump layers, representing a difference of 3.4%. In contrast, for the FreeSolv dataset, the impact of jump layers is more pronounced. The RMSE increases significantly from its lowest value of 1.042 at 4 jump layers to its highest value of 1.326 at 8 jump layers, a difference of 27%. The decline in performance at higher numbers

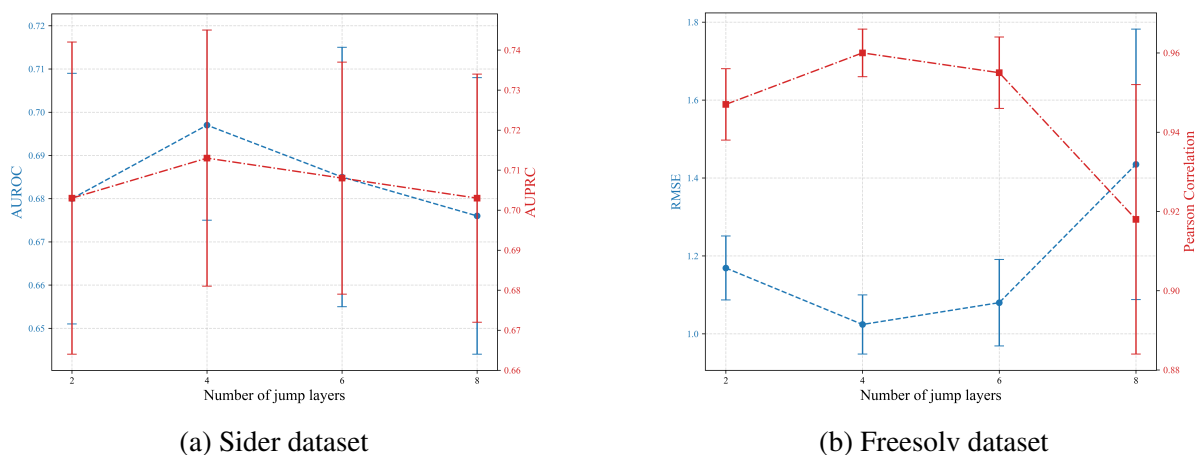


Figure 3.11: **Performance of the model with varying numbers of jump layers.** (a) Results on the SIDER dataset: the red line represents AUROC, and the blue line represents AUPRC. (b) Results on the FreeSolv dataset: the red line represents RMSE, and the blue line represents Pearson correlation.

of jump layers suggests that the inherent oversmoothing problem in GNNs may lead to the integration of overly smoothed deep features, which can negatively impact the performance of tasks requiring precise regression predictions.

3.5 Summary

In this study, we proposed a molecular representation learning framework that leveraged xLSTM for both atom-level and motif-level graphs, providing a novel approach to molecular property prediction. Additionally, we incorporated the MHMoE module into our framework, which dynamically assigned input features to diverse expert networks, enhancing predictive accuracy through fine-grained feature activation. The effectiveness of our model was demonstrated across 10 molecular property prediction datasets, showcasing its robust performance.

Our framework integrated atom-level and motif-level representations, and the ablation study highlighted the independent effectiveness of these two levels. Specifically, both the atom-level and motif-level networks achieved competitive results individually in classification tasks (section 3.4.4). However, the motif-level network exhibited a noticeable decline in regression performance. This limitation may have been due to the initialization features of the motif-level

graph, which relied on basic substructure properties, such as the counts of specific atoms (e.g., carbon) or bond types (e.g., single bonds). While these features captured useful information for classification tasks, they may have lacked the precision required for accurate regression predictions.

In addition to quantitative results, our interpretability analysis (section [3.4.5](#)) highlighted the strengths of the model. By analyzing the high-weight substructures identified by the model, we observed biologically meaningful correlations between the recognized substructures and specific molecular properties. This demonstrated that the model not only achieved competitive predictive performance but also provided valuable interpretability. Such interpretability was crucial for practical applications, as it could assist in drug design by guiding the identification of key molecular features associated with desired properties.

Chapter 4

DistRouting: Distribution-Aware Expert Routing Guided by Physicochemical Descriptors for Imbalanced Molecular Property Regression

Yan Sun, Yu Shi, Alana Deng, Carson Leung, Pingzhao Hu

Under review at NeurIPS 2025 (Submission ID:17299).

AUTHORS CONTRIBUTIONS

Conceptualization: Y.S., P.H.; Methodology: Y.S., Y.Shi, P.H.; Investigation: Y.S., Y.Shi, A.D.; Software: Y.S.; Validation: Y.S.; Visualization: Y.S.; Writing original draft preparation: Y.S.; Writing review and editing: Y.S., A.D., Y.Shi, C.L., P.H.; Supervision: C.L., P.H.; Funding acquisition: P.H. All authors have read and agreed to the submitted version of the manuscript.

4.1 Introduction

Accurately predicting molecular properties is a fundamental problem in computational chemistry and drug discovery [36, 48, 61]. Many molecular property prediction tasks are formulated as regression problems, where the goal is to estimate continuous-valued outcomes such as binding affinity, solubility, and toxicity. However, these tasks often suffer from long-tailed or imbalanced target distributions, where most data points concentrate in a narrow target range while chemically significant outliers are sparsely distributed. This distributional skew causes standard regression models to focus on dense target regions while neglecting rare but critical ones.

While deep learning has achieved remarkable progress in molecular property prediction, methods for addressing target imbalance in regression settings remain underexplored. Existing work on imbalanced learning largely centers on classification tasks, leveraging techniques such as resampling [138], cost-sensitive learning [139], and class-balanced loss [140]. For regression tasks with continuous-valued targets, prior efforts include both sample-level and model-level strategies. Synthetic Minority Over-sampling Technique for Regression with Gaussian Noise (SMOGN) [141] introduces oversampling by interpolating low-density regions. However, such methods may underperform on high-dimensional, structured data like molecular graphs, where naive interpolation can result in invalid or unrealistic samples that violate domain constraints [79, 142]. More recent approaches such as label and feature distribution smoothing (LDS/FDS) [143] adjust sample importance based on target density. Pu et al. [144] further propose group classification with descending soft labeling, discretizing targets and incorporating auxiliary supervision. While loss reweighting and sample-aware training have shown promise, few methods have explored architectural adaptations that specialize model behavior across different target regions. Notably, most of these methods originate from general-purpose regression research, rather than being developed for molecular property regression.

Inspired by the Mixture-of-Experts (MoE) paradigm [145], we aim to let each expert specialize in a different region of the target space. Mao et al. recently proposed KDEX, a

knowledge-diverse expert framework that trains specialized models for distinct class distributions [146]. However, their method is designed for classification tasks with discrete outputs and does not extend directly to regression.

We propose **DistRouting**: a distribution-aware expert routing framework for imbalanced molecular property regression. As illustrated in Figure 4.1a, DistRouting partitions the target space into intervals, with each expert dedicated to a specific interval. Beyond encoder-based gating, DistRouting incorporates additional similarity signals derived from RDKit-based physicochemical descriptors [102]. For each molecule, it computes similarity to expert centroids based on both the learned molecular embedding and the RDKit descriptor embedding. The combined similarity scores are transformed via softmax into a routing distribution over all experts, and the top- k with highest probabilities are selected. Descriptors such as topological polar surface area (TPSA) and hydrogen bonding provide chemically meaningful guidance, enhancing routing robustness in underrepresented target regions.

To align the learned routing behavior with the underlying target distribution, we derive soft labels from the distance between targets and interval centers, and apply a Kullback–Leibler divergence loss to encourage consistency between routing scores and these soft labels. Additionally, an interval-aware supervised contrastive learning loss is introduced to structure the embedding space, ensuring that molecules within the same interval form tighter clusters while remaining separated from other intervals.

Our main contributions are summarized as follows:

- We propose **DistRouting**, a distribution-aware expert routing framework based on the MoE paradigm for molecular property regression under imbalanced target distributions. It routes samples to specialized experts assigned to different target ranges, addressing the often-overlooked challenge of target imbalance in molecular property regression.
- We introduce a **physicochemical descriptors–guided routing mechanism**, which leverages RDKit-derived physicochemical descriptors to assist in expert selection. These features act as chemically informed priors that stabilize routing decisions.

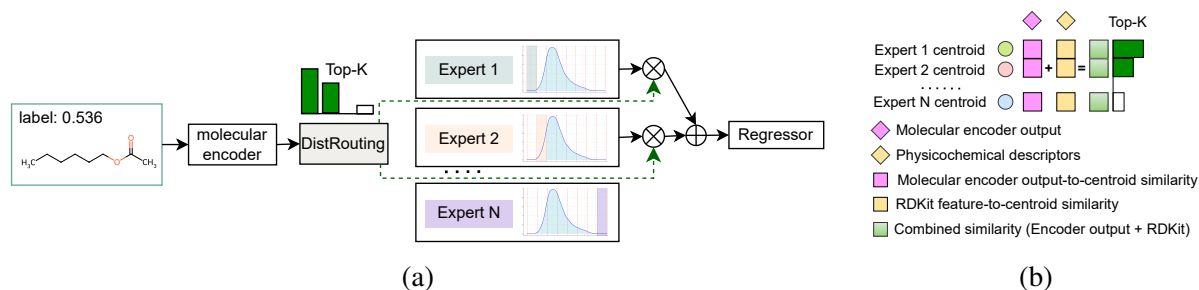


Figure 4.1: **DistRouting framework and routing mechanism.** (a) **Overall architecture.** DistRouting selects top- K experts based on similarity and aggregates their outputs for molecular property prediction. (b) **Routing mechanism.** Top- K selection is computed by combining similarity scores between expert centroids and both the molecular encoder output and RDKit-derived physicochemical descriptors.

- We further introduce an **interval-aware supervised contrastive learning loss** to structure the molecular representation space, promoting semantic alignment among samples within the same target interval to facilitate consistent expert routing.
- Extensive experiments show that integrating DistRouting into various molecular encoders consistently improves their performance, particularly in rare target regions.

4.2 Related Work

Molecular Encoders. Learning effective representations of molecules is fundamental to property prediction tasks. Traditional approaches rely on handcrafted descriptors or fingerprints, while recent advances adopt deep learning to learn representations directly from molecular structures. Graph Neural Networks (GNNs) have become a standard approach for molecular representation, where molecules are modeled as graphs with atoms as nodes and bonds as edges [48, 147, 148]. Various GNN architectures, such as Graph Attention Networks (GAT) [50] and DeeperGCN [90], improve message passing by incorporating attention mechanisms and residual connections.

In parallel, sequence-based encoders have gained popularity by processing Simplified Molecular Input Line Entry System (SMILES) strings as molecular language. Transformer-based

models such as ChemBERTa [77], SMILES-BERT [78], and Chemformer [149] leverage large-scale self-supervised pretraining on unlabeled SMILES data to capture structural patterns and chemical meaning. The resulting molecular representations serve as a foundation for downstream modules.

Mixture of Experts. Mixture-of-Experts (MoE) architectures [145, 150] dynamically route each input to a sparse subset of expert networks, enabling scalable and modular computation. A typical MoE layer consists of N experts, each implemented as an independent feed-forward network (FFN). Given an input representation u , the MoE output is computed as:

$$h = \sum_{i=1}^N g_i \cdot \text{FFN}_i(u), \quad (4.1)$$

$$g_i = \begin{cases} \text{Softmax}_i(s), & \text{if } i \in \text{TopK}(s, K), \\ 0, & \text{otherwise,} \end{cases} \quad (4.2)$$

$$s_i = u^\top e_i, \quad (4.3)$$

where s_i denotes the similarity score between input u and expert i , computed via dot product with a learnable centroid e_i . The softmax is applied over top- k entries of s , yielding a sparse gating vector g that activates only k experts per input.

Supervised Contrastive Learning. Contrastive learning is a powerful method for learning semantically meaningful representations by pulling similar samples together and pushing dissimilar ones apart [151]. Originally unsupervised, supervised contrastive learning [97] incorporates label information to define positive and negative pairs more effectively.

Formally, given a mini-batch of M samples with representations $\{z_i\}_{i=1}^M$ and corresponding class $\{y_i\}_{i=1}^M$, the supervised contrastive loss for anchor sample i is defined as:

$$\mathcal{L}_{\text{SupCon}} = \sum_{i=1}^M \frac{1}{|P(i)|} \sum_{j \in P(i)} -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k \in A(i)} \exp(\text{sim}(z_i, z_k)/\tau)}, \quad (4.4)$$

where $P(i)$ is the set of positive samples sharing the same label as i , $A(i)$ includes all samples

in the batch except i , $\text{sim}(\cdot, \cdot)$ denotes cosine similarity, and τ is a temperature parameter.

4.3 Method: DistRouting

4.3.1 Preliminaries and Notation

Let $\{(x_i, y_i)\}_{i=1}^N$ denote the training set, where $x_i \in \mathbb{R}^d$ is the input molecule and $y_i \in \mathbb{R}$ is the corresponding continuous-valued molecular property. We partition the label space $\mathcal{Y}$ into B intervals $\{I_1, \dots, I_B\}$, each defined by boundaries $[y_{b-1}, y_b]$, where $b \in \mathcal{B} = \{1, \dots, B\}$ indexes the intervals. Each interval I_b is associated with an interval center c_b , which is used for routing supervision and representation guidance. We assign a dedicated expert to each interval, enabling specialization across different target ranges. Given an input molecule x , we use a molecular encoder $f(x; \theta)$ to extract a representation $z \in \mathbb{R}^d$. Additionally, we extract 200 physicochemical descriptors from RDKit and map them to $\mathbb{R}^d$ via a multilayer perceptron (MLP) to obtain a descriptor embedding $r \in \mathbb{R}^d$, which is used to guide the expert routing process.

4.3.2 Distribution-aware Routing Guided by Physicochemical Descriptors

As illustrated in Figure 4.1b, we design a hybrid routing mechanism that leverages molecular embeddings and physicochemical descriptors to assign molecules to experts in a distribution-aware manner. Given a molecular embedding z and an RDKit-derived descriptor vector r , each expert, indexed by its interval $b \in \{1, \dots, B\}$, is associated with a learnable centroid vector $e_b \in \mathbb{R}^d$.

We compute the combined similarity between the input features and each expert centroid as:

$$s_b = z^\top e_b + r^\top e_b. \quad (4.5)$$

Routing weights are then computed by applying a softmax over the score vector $\{s_b\}$ and

retaining only the top- k experts:

$$g_b = \begin{cases} \text{softmax}_b(s), & \text{if } b \in \text{TopK}(s, K), \\ 0, & \text{otherwise,} \end{cases} \quad (4.6)$$

where $\text{softmax}_b(s)$ denotes the b -th component of the softmax applied over all scores $\{s_1, \dots, s_B\}$.

The final output is obtained by aggregating the responses from the selected experts:

$$z' = \sum_{b=1}^B g_b \cdot \text{FFN}_b(z), \quad (4.7)$$

where $\text{FFN}_b(\cdot)$ denotes the b -th expert network.

4.3.3 Gating Supervision with Soft Target Labels

As described in the problem setting, the continuous target space $\mathcal{Y}$ is partitioned into B intervals $\{I_1, I_2, \dots, I_B\}$, each with a centroid c_b and width w_b . This structure enables us to encode coarse-grained semantics over target values, which we leverage to supervise expert routing.

Given a sample (x_i, y_i) , we compute a soft target vector $\mathbf{q}_i \in \mathbb{R}^B$ indicating the degree to which the target y_i belongs to each interval. The assignment is based on a normalized Gaussian kernel, scaled by the width of each interval:

$$\mathbf{q}_{ib} = \frac{\exp\left(-\frac{1}{2} \left(\frac{y_i - c_b}{w_b \cdot \sigma}\right)^2\right)}{\sum_{j=1}^B \exp\left(-\frac{1}{2} \left(\frac{y_i - c_j}{w_j \cdot \sigma}\right)^2\right)}, \quad (4.8)$$

where σ is a temperature hyperparameter controlling the smoothness of the soft labels. The predicted routing distribution $\mathbf{p}_i \in \mathbb{R}^B$ is computed by applying softmax over the expert scores $\mathbf{s}_i = \{s_{i1}, \dots, s_{iB}\}$, where s_{ib} is the similarity score between sample i and expert b . The gating loss is then defined as:

$$\mathcal{L}_{\text{gate}} = \text{KL}(\mathbf{q}_i \| \mathbf{p}_i). \quad (4.9)$$

4.3.4 Interval-Aware Supervised Contrastive Learning

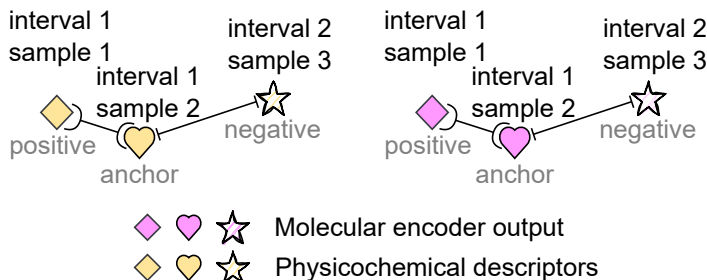


Figure 4.2: **ISCL mechanism.** ISCL constructs positives within the same interval and negatives across intervals, applied to both molecular embeddings and physicochemical descriptors.

To promote semantic structure in the representation space and encourage expert specialization, we introduce an interval-aware supervised contrastive learning (ISCL) loss. Unlike traditional contrastive learning that uses discrete class labels, ISCL is tailored to continuous regression targets by grouping molecules into intervals and computing soft target labels that reflect target proximity.

While the routing mechanism assigns molecules to experts based on feature similarity, ISCL shapes both the learned and physicochemical feature spaces by pulling together samples with similar soft labels and pushing apart those with dissimilar ones. As illustrated in Figure 4.2, ISCL constructs positive pairs from molecules within the same interval and negative pairs across different intervals, applied to both molecular embeddings and physicochemical descriptors.

Specifically, for each sample x_i with target y_i , we compute a soft target vector $\mathbf{q}_i \in \mathbb{R}^B$ using a normalized Gaussian kernel as described in Section 4.3.3. Given a minibatch of samples $\{(x_i, y_i)\}_{i=1}^M$, we extract two representations per sample: the molecular encoder output $z_i = f(x_i; \theta)$ and the corresponding RDKit-derived descriptor embedding $r_i \in \mathbb{R}^d$. ISCL is applied independently to both views, and the total contrastive loss is:

$$\mathcal{L}_{\text{ISCL}} = \mathcal{L}_{\text{ISCL}}^{\text{mol}} + \mathcal{L}_{\text{ISCL}}^{\text{rdkit}}, \quad (4.10)$$

Each component follows a weighted supervised contrastive formulation:

$$\mathcal{L}_{\text{ISCL}}^{(\cdot)} = -\frac{1}{|I|} \sum_{i \in I} \frac{1}{|P(i)|} \sum_{j \in P(i)} \log \frac{\exp(\text{sim}(v_i, v_j)/\tau)}{\sum_{k \neq i} \exp(\text{sim}(v_i, v_k)/\tau) \cdot w_{ik}}, \quad (4.11)$$

where v_i is either z_i or r_i , and the weighting term is defined as:

$$w_{ik} = \gamma^{1 - \cos(\mathbf{q}_i, \mathbf{q}_k)}, \quad (4.12)$$

with $\gamma > 1$ controlling the penalty strength and where $\cos(\cdot)$ denotes cosine similarity. This encourages stronger repulsion between samples from dissimilar target intervals.

4.3.5 Prediction and Overall Loss

Following expert routing and aggregation, the fused representation is passed through a MLP head to generate the final prediction:

$$\hat{y} = \text{MLP}(z'), \quad (4.13)$$

where z' is the output from the sparse expert layer described in Eq. (4.7). The overall training objective combines three components:

$$\mathcal{L} = \mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{gate}} + \lambda \cdot \mathcal{L}_{\text{ISCL}}, \quad (4.14)$$

where $\mathcal{L}_{\text{reg}}$ is the mean squared error loss, $\mathcal{L}_{\text{gate}}$ is the KL divergence loss supervising expert routing (Section 4.3.3), and $\mathcal{L}_{\text{ISCL}}$ is the interval-aware supervised contrastive loss (Section 4.3.4). The coefficient λ balances the contrastive objective. A full training procedure is provided in Algorithm 1.

Algorithm 1: DistRouting: Distribution-Aware Expert Routing with Physicochemical Guidance

Input: Training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$; Molecular encoder $f(\cdot; \theta)$; RDKit extractor $\text{rdk}(\cdot)$; RDKit encoder $\text{enc}_{\text{rdk}}(\cdot)$; Number of experts B ; top- k routing k ; Temperature parameters τ, σ ; ISCL weight λ

Output: Trained encoder $f(\cdot)$ and expert networks $\{\text{FFN}_j\}_{j=1}^B$

```

1 Partition  $\mathcal{Y}$  into  $B$  intervals with  $\{c_j\}$  and  $\{w_j\}$ ;
2 Initialize expert centroids  $\{e_j\}_{j=1}^B$ ;
3 while not converged do
4   Sample minibatch  $\{(x_i, y_i)\}_{i=1}^M$ ;
5   foreach  $x_i$  in minibatch do
6      $z_i \leftarrow f(x_i)$ ;  $r_i \leftarrow \text{enc}_{\text{rdk}}(\text{rdk}(x_i))$ ;
7     foreach expert  $j = 1$  to  $B$  do
8        $s_{ij} \leftarrow \text{sim}(z_i, e_j) + \text{sim}(r_i, e_j)$ ;
9      $\alpha_{ij} \leftarrow \text{softmax}_j(s_{ij})$ ;
10     $g_{ij} \leftarrow \alpha_{ij}$  if  $j \in \text{TopK}(\alpha_i, k)$  else 0;
11     $h_i \leftarrow \sum_{j=1}^B g_{ij} \cdot \text{FFN}_j(z_i)$ ;
12     $\hat{y}_i \leftarrow \text{MLP}(h_i)$ ;
13    Compute  $w_{ij} = \frac{\exp\left(-\frac{1}{2}\left(\frac{y_i - c_j}{w_j \sigma}\right)^2\right)}{\sum_{l=1}^B \exp\left(-\frac{1}{2}\left(\frac{y_i - c_l}{w_l \sigma}\right)^2\right)}$ ;
14     $\mathcal{L}_{\text{reg}} \leftarrow \text{MSE}(\hat{y}_i, y_i)$ ;
15     $\mathcal{L}_{\text{gate}} \leftarrow \text{KL}(w_i \| \alpha_i)$ ;
16     $\mathcal{L}_{\text{ISCL}}$  from Eq. (5);
17     $\mathcal{L} \leftarrow \mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{gate}} + \lambda \mathcal{L}_{\text{ISCL}}$ ;
18    Update parameters;

```

4.4 Experiments

4.4.1 Experimental Setting

Datasets. We evaluate DistRouting on four molecular property regression benchmarks from the Therapeutics Data Commons (TDC) [131]: **Caco2_Wang**, **Lipophilicity_AstraZeneca**, **PPBR_AZ**, and **LD50_Zhu**. These datasets span a range of biophysical and pharmacokinetic properties, as summarized in Table 4.1. Caco2 and PPBR contain fewer than 1,000 samples,

Table 4.1: Overview of the datasets.

Dataset	#Samples	Unit	Range (Y)	Description
Caco2	906	logPapp	[-7.5, -3.6]	Caco-2 cell permeability; simulates intestinal absorption.
PPBR	1,000	%	[7.8, 100]	Plasma protein binding ratio; reflects drug availability in blood.
Lipo	4,200	logD	[-1.3, 4.5]	Lipophilicity; influences drug absorption and distribution.
LD50	7,385	mg/kg	[0.01, 7.1]	Median lethal dose; widely used measure of acute toxicity.

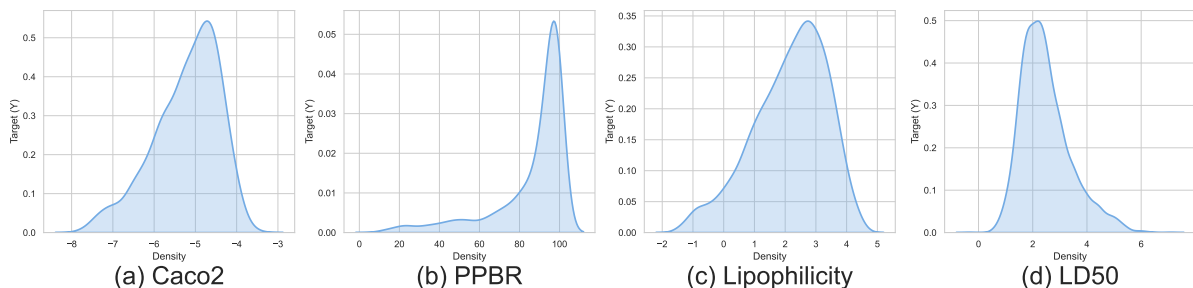


Figure 4.3: Target distributions of the datasets.

while Lipo and LD50 are substantially larger, with over 4,000 and 7,000 compounds respectively. The prediction tasks cover a diverse set of pharmacokinetic and toxicological endpoints. Figure 4.3 illustrates the target distributions of each dataset. All four exhibit varying degrees of imbalance, with PPBR and LD50 showing long-tailed or skewed patterns, while Caco2 and Lipo have more compact but unevenly sampled target ranges. These distributional characteristics pose challenges for regression models, particularly in underrepresented regions of the target space. We follow the standard 5-fold scaffold split provided by TDC, which partitions molecules based on their Bemis–Murcko scaffolds to prevent information leakage from similar chemical structures. Each fold consists of a 70%/10%/20% split into training, validation, and test sets, respectively.

Baselines. To assess the general applicability of DistRouting under the underexplored setting of molecular property regression with target imbalance, we integrate it as a plug-in module into three representative backbone encoders covering both graph- and SMILES-based representations: GAT [50], DeeperGCN [90], and ChemBERTa [77]. For each backbone, we compare the original model with its DistRouting-enhanced variant to evaluate the isolated effect of our method.

The GAT baseline consists of a 4-layer Graph Attention Network with 512 hidden units per layer, ReLU activation, and a dropout rate of 0.2. It employs Jumping Knowledge (JK) via concatenation for multi-level feature aggregation, without normalization layers.

DeeperGCN is implemented as a 4-layer graph convolutional network using GENConv blocks. Each layer adopts residual connections, PReLU activation, and batch normalization, with 512 hidden dimensions. Global mean pooling is used for graph-level readout.

ChemBERTa is a transformer-based encoder pretrained on SMILES sequences. We utilize the DeepChem/ChemBERTa-77M-MLM variant, which consists of 6 transformer layers, 384 hidden dimensions, and 12 attention heads. The final molecular representation is obtained from the [CLS] token embedding.

The output representation from the DistRouting module is passed to a two-layer feedforward regressor with a ReLU activation in between. The regressor maps from the input feature dimension to a hidden dimension (512 in our experiments), and finally outputs a scalar property prediction.

Each expert is a two-layer feedforward network. The input is first projected to a lower expert-specific hidden dimension, followed by ReLU activation and a second linear transformation back to the original size. Experts are initialized with Xavier initialization. Only the top- k experts selected by the router contribute to each prediction.

Evaluation Process and Details. We evaluate model performance using both mean absolute error (MAE) and Pearson correlation coefficient (PCC) to assess prediction accuracy and correlation strength, respectively. To further examine robustness under target imbalance, we report region-wise MAE and PCC on the head and tail intervals, defined as the bottom and top 20% quantiles of the target distribution. All results are reported with mean and standard deviation over 5 splits. The target space is partitioned into $B = 8$ intervals, and for each sample, the top- $k = 2$ most relevant experts are selected via similarity-guided routing. The soft label smoothing parameter σ in Eq. 4.8 is set to 0.7. All models are trained using the Adam optimizer with a learning rate of $1e-4$ and a batch size of 128, with the best model selected

based on validation MAE.

Table 4.2: Hyperparameter settings.

Dataset	λ	γ
Caco2	0.1	4
PPBR	0.1	4
Lipophilicity	1.0	10
LD50	1.0	10

Table 4.2 summarizes the dataset-specific hyperparameters used in DistRouting, including the ISCL loss weight λ (Eq. (4.14)) and the repulsion strength parameter γ (Eq. (4.12)), which controls the penalty for dissimilar sample pairs. These hyperparameters were selected via grid search over $\lambda \in \{0.1, 1.0\}$ and $\gamma \in \{2, 4, 10, 20, 50\}$.

4.4.2 Overall Performance

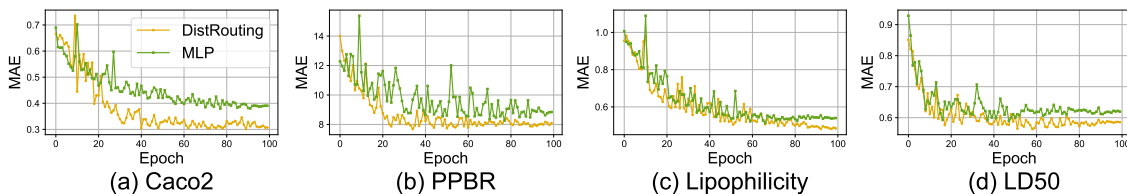


Figure 4.4: **MAE on the validation set over training epochs.** Across all four regression tasks, DistRouting (yellow) converges faster and reaches lower error compared to the vanilla encoder (green).

Table 4.3: MAE ($\downarrow$) on the four datasets. Bold numbers indicate the best result in each column.

Method	Caco2	Lipophilicity	PPBR	LD50
DeeperGCN	0.366 $\pm$ 0.012	0.528 $\pm$ 0.012	8.355 $\pm$ 0.211	0.643 $\pm$ 0.010
DeeperGCN + DistRouting	0.315 $\pm$ 0.009	0.509 $\pm$ 0.013	7.849 $\pm$ 0.028	0.616 $\pm$ 0.009
GAT	0.383 $\pm$ 0.012	0.607 $\pm$ 0.008	7.940 $\pm$ 0.148	0.651 $\pm$ 0.014
GAT + DistRouting	0.327 $\pm$ 0.011	0.551 $\pm$ 0.010	7.780 $\pm$ 0.154	0.609 $\pm$ 0.026
ChemBERTa	0.421 $\pm$ 0.011	0.603 $\pm$ 0.007	8.714 $\pm$ 0.200	0.672 $\pm$ 0.007
ChemBERTa + DistRouting	0.410 $\pm$ 0.024	0.592 $\pm$ 0.010	8.517 $\pm$ 0.218	0.645 $\pm$ 0.020

Table 4.4: PCC ($\uparrow$) on the four datasets. Bold numbers indicate the best result in each column.

Method	Caco2	Lipophilicity	PPBR	LD50
DeeperGCN	0.788 ± 0.010	0.810 ± 0.007	0.552 ± 0.021	0.568 ± 0.005
DeeperGCN + DistRouting	0.832 ± 0.014	0.826 ± 0.007	0.623 ± 0.009	0.607 ± 0.024
GAT	0.794 ± 0.022	0.767 ± 0.007	0.618 ± 0.013	0.540 ± 0.027
GAT + DistRouting	0.819 ± 0.015	0.794 ± 0.004	0.624 ± 0.013	0.613 ± 0.042
ChemBERTa	0.724 ± 0.024	0.768 ± 0.003	0.466 ± 0.022	0.504 ± 0.021
ChemBERTa + DistRouting	0.762 ± 0.016	0.781 ± 0.007	0.520 ± 0.007	0.546 ± 0.029

Figure 4.4 presents the MAE curves on the validation sets across training epochs for the four datasets. DistRouting not only converges faster than the MLP baseline but also consistently reaches lower MAE, indicating more stable and effective learning dynamics. These improvements are consistent across datasets and model backbones.

Tables 4.3 and 4.4 report the overall results in terms of MAE ($\downarrow$) and PCC ($\uparrow$) on the test sets. For each backbone, we compare the vanilla model with its DistRouting-augmented variant. We observe consistent improvements in both MAE and PCC across all datasets and backbones with the integration of DistRouting.

DeeperGCN performs strongly overall, with a 13.9% reduction in MAE on Caco2 (from 0.366 to 0.315), a 6.1% reduction on PPBR (from 8.355 to 7.849). Its PCC improves on PPBR (+0.071, from 0.552 to 0.623) and LD50 (+0.039, from 0.568 to 0.607). For **GAT**, notable MAE reductions are observed on Caco2 (14.6%, from 0.383 to 0.327) and Lipophilicity (9.2%, from 0.607 to 0.551). A substantial PCC gain is also seen on LD50 (+0.073, from 0.540 to 0.613). While **ChemBERTa** shows relatively smaller MAE gains compared to the other two backbones, it still achieves a meaningful reduction on LD50 (4.02%, from 0.672 to 0.645). Its PCC also improves on PPBR (+0.054, from 0.466 to 0.520). The consistent PCC improvements suggest that DistRouting enhances not only numerical accuracy but also the biological relevance of model predictions by better aligning with property trends.

Table 4.5: Region-wise MAE ($\downarrow$) across all datasets and methods. Best between each pair is bolded.

Dataset	Method	Head MAE $\downarrow$	Body MAE $\downarrow$	Tail MAE $\downarrow$
Caco2	DeeperGCN	0.359 ± 0.031	0.376 ± 0.015	0.354 ± 0.041
	DeeperGCN + DistRouting	0.306 ± 0.025	0.341 ± 0.032	0.267 ± 0.039
	GAT	0.524 ± 0.031	0.298 ± 0.009	0.350 ± 0.054
	GAT + DistRouting	0.285 ± 0.039	0.358 ± 0.041	0.323 ± 0.030
	ChemBERTa	0.402 ± 0.036	0.456 ± 0.025	0.366 ± 0.023
	ChemBERTa + DistRouting	0.454 ± 0.032	0.415 ± 0.043	0.321 ± 0.024
Lipophilicity	DeeperGCN	0.656 ± 0.029	0.449 ± 0.012	0.638 ± 0.037
	DeeperGCN + DistRouting	0.597 ± 0.022	0.445 ± 0.015	0.614 ± 0.022
	GAT	0.881 ± 0.035	0.457 ± 0.013	0.782 ± 0.047
	GAT + DistRouting	0.673 ± 0.032	0.464 ± 0.006	0.691 ± 0.064
	ChemBERTa	0.737 ± 0.023	0.542 ± 0.008	0.653 ± 0.024
	ChemBERTa + DistRouting	0.727 ± 0.015	0.532 ± 0.013	0.637 ± 0.054
PPBR	DeeperGCN	19.671 ± 0.398	6.041 ± 0.282	3.923 ± 0.436
	DeeperGCN + DistRouting	19.285 ± 0.425	5.633 ± 0.095	2.992 ± 0.232
	GAT	18.541 ± 0.997	5.591 ± 0.431	4.346 ± 0.341
	GAT + DistRouting	19.142 ± 1.801	5.722 ± 0.575	2.515 ± 0.368
	ChemBERTa	22.094 ± 1.195	5.750 ± 0.511	4.170 ± 0.610
	ChemBERTa + DistRouting	20.152 ± 1.248	6.469 ± 0.562	2.945 ± 0.366
LD50	DeeperGCN	0.513 ± 0.033	0.454 ± 0.022	1.338 ± 0.053
	DeeperGCN + DistRouting	0.500 ± 0.019	0.465 ± 0.024	1.182 ± 0.062
	GAT	0.495 ± 0.028	0.445 ± 0.004	1.423 ± 0.074
	GAT + DistRouting	0.471 ± 0.020	0.438 ± 0.017	1.258 ± 0.153
	ChemBERTa	0.465 ± 0.011	0.455 ± 0.013	1.527 ± 0.059
	ChemBERTa + DistRouting	0.474 ± 0.024	0.453 ± 0.016	1.391 ± 0.080

4.4.3 Region-wise Performance

Table 4.5 and Figure 4.5 present MAE performance in the **head**, **body**, and **tail** regions across all datasets, comparing each backbone model with its DistRouting-enhanced version. DistRouting consistently improves MAE in the **tail** region across all datasets and backbones, with particularly notable gains on LD50, which exhibits the most severe target imbalance in the tail. DeeperGCN achieves an 11.7% reduction in tail MAE on LD50 (from 1.338 to 1.182), while GAT achieves a comparable 11.6% reduction (from 1.423 to 1.258). Although Chem-

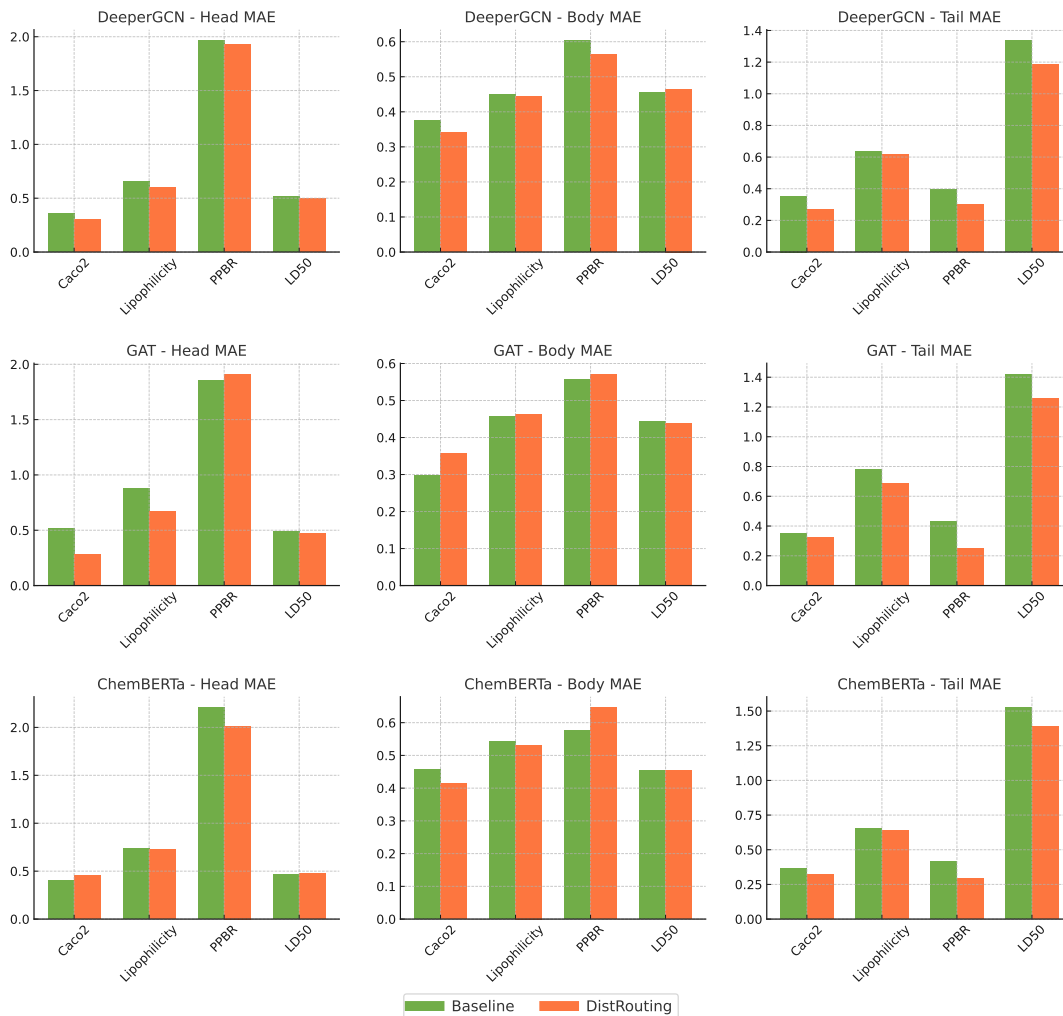


Figure 4.5: Region-wise MAE across four datasets and three backbone models. Red bars denote results with DistRouting, while green bars correspond to the baseline models without distribution-enhanced routing. Note that PPBR values have been scaled down to allow visual comparison.

BERTa shows relatively smaller improvements, it still achieves an 8.9% reduction in tail MAE on LD50 (from 1.527 to 1.391).

In the **head** region, DistRouting also brings improvements in most settings. According to the target distribution, PPBR exhibits noticeable sparsity in this segment. Accordingly, we observe a reduction in head MAE for ChemBERTa on PPBR (from 22.094 to 20.152). Furthermore, as shown in Table 4.6, DistRouting consistently improves PCC in the head region of PPBR across all backbones: from 0.243 to 0.406 for DeeperGCN, from 0.333 to 0.397 for

Table 4.6: Region-wise PCC ($\uparrow$) across all datasets and backbones. Bold indicates better performance between paired methods.

Dataset	Method	Head PCC $\uparrow$	Body PCC $\uparrow$	Tail PCC $\uparrow$
Caco2	DeeperGCN	0.686 ± 0.035	0.381 ± 0.050	0.312 ± 0.069
	DeeperGCN + DistRouting	0.789 ± 0.045	0.446 ± 0.058	0.386 ± 0.048
	GAT	0.771 ± 0.016	0.346 ± 0.040	0.334 ± 0.028
	GAT + DistRouting	0.757 ± 0.022	0.401 ± 0.048	0.370 ± 0.045
	ChemBERTa	0.501 ± 0.059	0.276 ± 0.055	0.251 ± 0.059
	ChemBERTa + DistRouting	0.463 ± 0.033	0.351 ± 0.044	0.180 ± 0.044
Lipophilicity	DeeperGCN	0.593 ± 0.018	0.565 ± 0.021	0.269 ± 0.027
	DeeperGCN + DistRouting	0.625 ± 0.019	0.589 ± 0.010	0.260 ± 0.035
	GAT	0.555 ± 0.014	0.493 ± 0.017	0.287 ± 0.011
	GAT + DistRouting	0.558 ± 0.016	0.567 ± 0.010	0.138 ± 0.047
	ChemBERTa	0.556 ± 0.018	0.498 ± 0.003	0.108 ± 0.032
	ChemBERTa + DistRouting	0.555 ± 0.028	0.509 ± 0.014	0.108 ± 0.021
PPBR	DeeperGCN	0.243 ± 0.068	0.313 ± 0.032	0.232 ± 0.022
	DeeperGCN + DistRouting	0.406 ± 0.053	0.342 ± 0.023	0.271 ± 0.046
	GAT	0.333 ± 0.036	0.306 ± 0.010	0.273 ± 0.040
	GAT + DistRouting	0.397 ± 0.043	0.292 ± 0.036	0.243 ± 0.033
	ChemBERTa	0.221 ± 0.046	0.264 ± 0.030	0.219 ± 0.037
	ChemBERTa + DistRouting	0.302 ± 0.025	0.321 ± 0.013	0.252 ± 0.015
LD50	DeeperGCN	0.165 ± 0.039	0.369 ± 0.012	0.061 ± 0.059
	DeeperGCN + DistRouting	0.166 ± 0.034	0.392 ± 0.012	0.116 ± 0.075
	GAT	0.196 ± 0.034	0.350 ± 0.021	0.039 ± 0.038
	GAT + DistRouting	0.155 ± 0.039	0.397 ± 0.031	0.137 ± 0.041
	ChemBERTa	0.204 ± 0.032	0.375 ± 0.017	0.053 ± 0.025
	ChemBERTa + DistRouting	0.194 ± 0.050	0.406 ± 0.018	0.056 ± 0.028

GAT, and from 0.221 to 0.302 for ChemBERTa.

The **body** region remains relatively stable across models, with DistRouting showing modest improvements or parity in most cases. This is expected, as the body contains the densest regions of the target distribution and is less susceptible to imbalance-induced degradation. These results underscore the strength of DistRouting in enhancing generalization across different regions of the target space, particularly in underrepresented areas where vanilla models often struggle to perform reliably.

Table 4.7: Ablation study of DistRouting components. Each row corresponds to a variant with specific modules removed. ✓ indicates the component is used. We report MAE (↓) on four datasets.

Variant	MoE Routing	RDKit	Gate Sup.	ISCL	Caco2 ↓	PPBR ↓	Lipo ↓	LD50 ↓
Full Model	✓	✓	✓	✓	0.315	7.849	0.509	0.616
w/o Gating Sup.	✓	✓	✗	✓	0.343	8.145	0.510	0.632
w/o ISCL	✓	✓	✓	✗	0.322	7.859	0.523	0.608
w/o RDKit Guidance	✓	✗	✓	✓	0.340	8.266	0.507	0.637
MoE Routing only	✓	✗	✗	✗	0.343	8.314	0.530	0.641

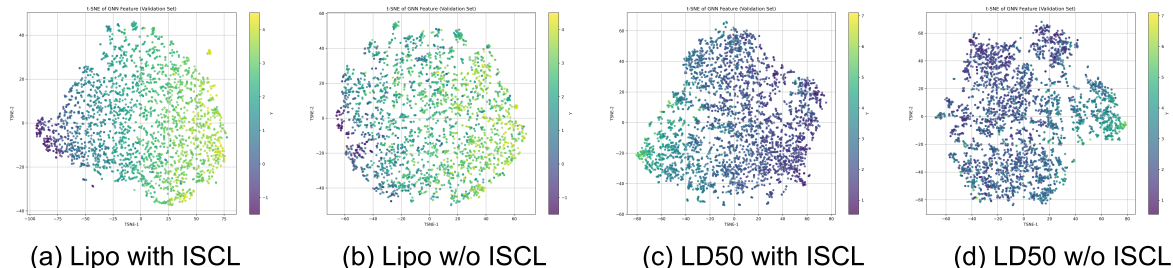


Figure 4.6: **t-SNE visualization of molecular representations.** Molecular embeddings for the Lipophilicity and LD50 datasets under two settings: with ISCL (a, c) and without ISCL (b, d). Representations are extracted after the molecular encoder and before expert routing.

4.4.4 Ablation Study

Ablation analysis of DistRouting components. Based on the ablation results in Table 4.7, conducted with DeeperGCN as the backbone, we find that gating supervision is the most critical component: its removal consistently degrades performance across all datasets. This underscores the role of soft target guidance in aligning expert assignments with the semantic structure of the target space.

The impact of ISCL, in contrast, is less consistent. While it brings clear improvements on the Lipophilicity dataset, its effect is minimal on others and even slightly detrimental on LD50. Nevertheless, t-SNE visualizations in Figure 4.6 suggest that ISCL meaningfully enhances the structure of the representation space, producing more organized embeddings.

We further investigate this discrepancy by varying the ISCL loss weight, with results summarized in Table 4.8. We found that performance decreases when ISCL is applied with a small weight, but partially recovers as the weight increases on the LD50 dataset. Notably, when λ is

Table 4.8: Validation MAE under different ISCL weights λ .

λ	Caco2	PPBR	Lipo	LD50
0.1	0.315	7.849	0.508	0.643
1.0	0.326	8.427	0.509	0.616
2.0	0.345	8.403	0.509	0.624
3.0	0.377	8.267	0.511	0.605

set to 3, the MAE on LD50 improves to 0.605. These observations suggest that when ISCL is applied with a small weight, it may be insufficient to guide representation learning effectively, and might even introduce conflicting signals relative to the gating objective on LD50.

Removing RDKit guidance also leads to moderate performance drops, though the model still outperforms the baseline encoder. This demonstrates that even without the guidance of physicochemical descriptors, the routing mechanism retains meaningful inductive bias through learned representations alone. Finally, the MoE-only variant, which disables all auxiliary supervision, performs the worst overall. This underscores that naively routing to experts without guidance is ineffective and confirms the necessity of distribution-aware mechanisms for expert specialization.

4.5 Discussion

Table 4.9: JS distance ($\downarrow$) between predicted and true target distributions.

Method	Caco2	PPBR	LD50	Lipo
DeeperGCN	0.1530	0.1417	0.1872	0.1508
+ DistRouting	0.1265	0.1233	0.1444	0.1342

Distribution Matching. To evaluate the similarity between the predicted and true target distributions, we compute the Jensen–Shannon (JS) distance, which is the square root of the Jensen–Shannon divergence—a symmetric and smoothed variant of the Kullback–Leibler (KL) divergence. Given two probability distributions P and Q over the same discrete support, the JS

distance is defined as:

$$\text{JSD}(P \parallel Q) = \sqrt{\frac{1}{2}D_{\text{KL}}(P \parallel M) + \frac{1}{2}D_{\text{KL}}(Q \parallel M)} \quad \text{where} \quad M = \frac{1}{2}(P + Q)$$

Here, $\text{KL}(\cdot \parallel \cdot)$ denotes the Kullback–Leibler divergence. A smaller JS distance indicates greater similarity between the two distributions, with a value of zero signifying identical distributions.

As shown in Table 4.9, when implemented on the DeeperGCN backbone, DistRouting achieves lower JS distances than the baseline across all four datasets, indicating better global alignment with the true label distribution. The improvements are especially notable on datasets with pronounced target skew, such as LD50 (from 0.1872 to 0.1444).

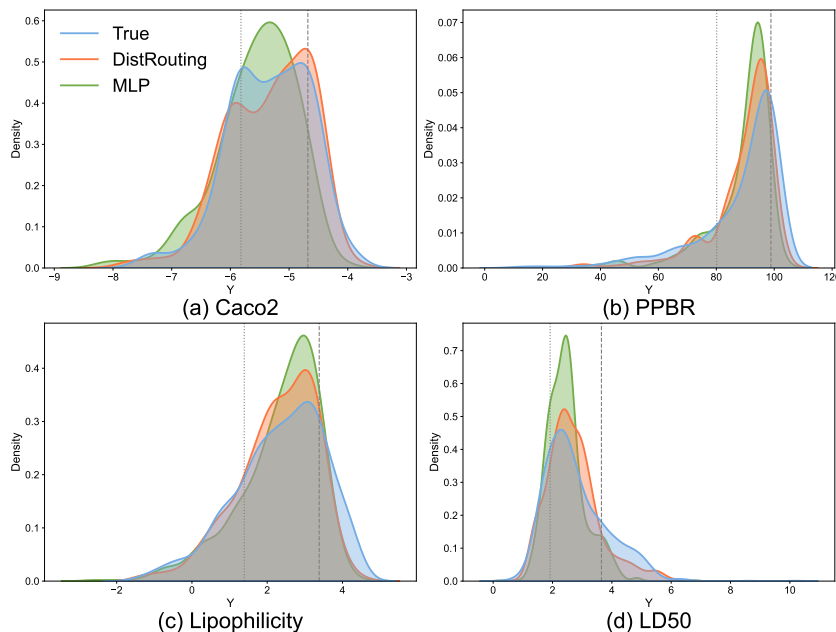


Figure 4.7: Distribution matching between predicted and true targets across datasets. The plot compares DistRouting (red), baseline encoder (green), and the true target distribution (blue).

Figure 4.7 shows how the baseline encoder and its DistRouting-enhanced version align with the true value distributions on four test datasets. In all cases, the baseline encoder exhibits a strong central bias, with predictions overly concentrated around the most frequent target values. This results in undercoverage of both the head and tail regions. By contrast, DistRouting provides a more faithful approximation of the target distribution by better reflecting asymme-

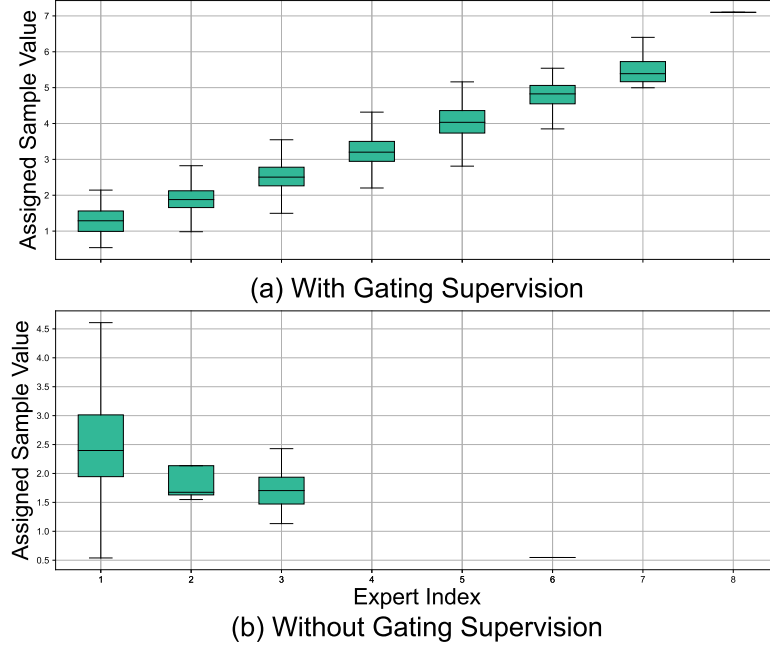


Figure 4.8: Distribution of expert assignments across target values on the LD50 dataset: (a) with gating supervision; (b) without gating supervision.

tries and capturing variations in head and tail regions. These qualitative patterns are consistent across datasets, reinforcing the quantitative findings in Table 4.5, where DistRouting achieves lower MAE in both head and tail regions.

Gating Behavior Analysis. To assess the influence of gating supervision, Figure 4.8 compares the distribution of expert assignments across target values with and without KL-based gating supervision on the LD50 dataset. In the supervised setting (Figure 4.8a), each expert learns to specialize in a distinct region of the target space, forming a structured partition aligned with interval semantics. In contrast, when gating supervision is removed (Figure 4.8b), the routing behavior becomes disorganized: multiple experts collapse to overlapping or narrow value ranges, while some are effectively unused.

This comparison reveals that the gating supervision plays a critical role in promoting expert diversity and enforcing consistent expert-target alignment. Without this loss, the model struggles to utilize expert capacity effectively. These findings align with the ablation results in Table 4.7, where removing gating supervision leads to a significant performance drop.

4.6 Summary

In this work, we presented **DistRouting**, a distribution-aware expert routing framework built on the Mixture-of-Experts paradigm to improve molecular property regression under imbalanced target distributions. DistRouting assigned samples to specialized experts aligned with specific target ranges. To enhance expert assignment, we introduced a routing mechanism guided by RDKit-derived physicochemical descriptors, incorporating chemically meaningful priors to improve routing stability. We also proposed an interval-aware supervised contrastive learning (ISCL) loss to structure the molecular representation space. Integrated into various molecular encoders, DistRouting consistently improved performance, particularly in rare target regions. This contribution helped fill an important gap by addressing the challenge of target imbalance, which remained underexplored in previous studies.

Limitations. While DistRouting showed strong empirical performance, it relied on uniform interval partitioning of the target space. Future work could explore data-driven or adaptive strategies to better reflect target density. Additionally, the interaction between ISCL and expert routing was sensitive to hyperparameter settings and might benefit from more robust coordination mechanisms.

Chapter 5

iNGNN-DTI: Prediction of Drug - Target Interaction with Interpretable Nested Graph Neural Network and Pretrained Molecule Models

Yan Sun, Yan Yi Li, Carson K. Leung, Pingzhao Hu

Bioinformatics, 2024, 40(3), btae135

DOI: [10.1093/bioinformatics/btae135](https://doi.org/10.1093/bioinformatics/btae135)

Reprinted with permission from Oxford University Press.

AUTHORS CONTRIBUTIONS

Conceptualization: Y.S., P.H.; Methodology: Y.S., Y.Y.L., P.H.; Investigation: Y.S., Y.Y.L.; Data curation: Y.S.; Software: Y.S.; Visualization: Y.S.; Writing original draft preparation: Y.S.; Writing review and editing: Y.S., Y.Y.L., C.K.L., P.H.; Supervision: C.K.L., P.H.; Funding acquisition: P.H. All authors have read and approved the final version of the manuscript.

5.1 Introduction

Drugs typically refer to small molecules, while targets often refer to macromolecules such as proteins. A drug may change the function of a biological target when it binds to the target, which is known as drug–target interaction (DTI) [152]. The prediction of DTI plays a key role in drug discovery. Since it is time-consuming and expensive to identify DTI pairs through biological assays, many computer-assisted methods have been developed [153]. With the success of deep learning in various areas, deep learning-based DTI prediction methods have exploded in recent years [154].

The base framework of deep learning methods for DTI involves encoding molecules and targets separately through two branches. Subsequently, the output features from these two branches are concatenated and fed into a classifier constructed using a fully connected network. Various deep learning models are utilized for feature representation learning when different raw representations for drugs and proteins are employed [37, 47, 155, 156]. For example, DeepDTA [37] employs a three-layer convolutional neural network (CNN) on the simplified molecular-input line-entry system (SMILES) string of drug molecules and the amino acid sequence of protein targets. In contrast, MolTrans [156] utilizes the transformer encoder, a module known for its excellent performance in natural language processing (NLP) tasks, on both sequence data.

However, the sequence data loses the structural information of the molecule. To address this limitation, approaches that directly learn representations from the molecular graph have been developed. The graph, constructed by nodes representing atoms and edges representing bonds, offers a more natural representation of molecules. Various graph neural networks (GNNs), such as graph convolutional networks (GCN) and graph attention networks (GAT), have been implemented in DTI tasks [64, 157]. A deeper review of different deep learning approaches for compound–protein interaction prediction can be found in [157]. However, most methods focus on encoding drugs with GNNs, with limited attempts to encode protein targets using GNNs. The challenge in representing proteins as graphs arises from the fact that the structures of most

proteins are unknown. The recent breakthrough achieved by AlphaFold2 [46] in predicting protein structures has eased this challenge, allowing the incorporation of graph data for proteins in DTI tasks. Our approach involves generating protein structures with AlphaFold2 and applying GNNs to both drugs and targets.

Interpretability is another challenge for the DTI task. Most methods learn representations of drugs and proteins with deep learning models, but they lack interpretation of the interaction. As is known, interactions happen between the substructures of drugs and targets [158]. ML-DTI [155] adds a mutual learning module between the drug encoder and the target encoder, which calculates attention on each atom of the drug. MolTrans [156] designs a pairwise interaction module after the encoder. It decomposes drugs and proteins into substructures and calculates interactions between every two substructures. DrugBAN [159] develops a novel bilinear interaction module to capture local structural interactions. These methods are all realized using protein sequence data. However, the sequence data loses detailed local structural information, which is important to binding site modeling.

Furthermore, the number of molecules in current DTI datasets is tiny compared to the scale of chemical space. This data sparsity may limit the generalization ability of DTI models to new drugs or targets that are unseen during training.

To address the above-mentioned limitations, we propose an interpretable nested graph neural network for drug–target interaction prediction (iNGNN-DTI), based on the graph data of both drugs and proteins. We adopt the nested graph neural network (NGNN) [160] as the fundamental feature extraction model to enhance substructure representation. This is because the representation of each node in NGNN is the result of pooling the k -hop subgraph surrounding it. After the encoder, we use a cross-attention-free transformer (cross-AFT) [161] module to calculate the interaction between each pair of nodes from drugs and targets.

In addition, we integrate features learned from two large-scale pre-trained models: Chemformer [149] for drugs and ESM [74] for proteins, and combine them with the NGNN outputs to further enrich the feature representations.

The contributions of our work include: 1) the proposed method utilizes the k-subgraph GNN extractor layer from the NGNN to encode the graph data. This particular layer has demonstrated superior performance compared to a standard GNN layer; 2) the method proposes to enhance performance by integrating the features of targets and drugs that have been extracted using models pre-trained on other large-scale unlabeled datasets; 3) we propose an interaction module on the graph neural network for the DTI task, and the experiment results indicate that it performs better than the vanilla GNN.

5.2 Materials and Methods

5.2.1 Datasets

We evaluate our method on three popularly used benchmark datasets: Davis [43], KIBA [29], and BIOSNAP [162]. The Davis and KIBA datasets are both for the kinase protein family. The Davis dataset indicates the activity of the drug with the equilibrium dissociation constant K_d value. The KIBA dataset combines dissociation constant K_d , inhibitory constant K_i , and the half-maximal inhibitory concentration IC_{50} .

These two datasets need preprocessing to generate binary labels, following the analysis in previous works [29, 156]. For the Davis dataset, the K_d value is transformed into log space as pK_d :

$$pK_d = -\log_{10}\left(\frac{K_d}{1e9}\right) \quad (5.1)$$

We set samples with $pK_d \geq 7$ as binding samples. For the KIBA dataset, the threshold is 12.1, and samples with KIBA scores larger than the threshold are set as positive. BIOSNAP collects high-throughput experiment results from diverse resources, only providing positive pairs. We randomly sample the same number of negative pairs. The statistics of the datasets are summarized in Table 5.1.

Table 5.1: Summary of the datasets

Dataset	#Drugs	#Proteins	#Interactions	#Positives/#Negatives
Davis	68	442	30056	1506 / 28550
KIBA	2068	229	118254	22729 / 95525
BIOSNAP	4510	2180	27428	13817 / 13611

5.2.2 Input Data Representation

Both drugs and targets can be represented in two different forms: sequence data and graph data. This section outlines the methods used to construct graph inputs for drugs and targets using their respective sequence data.

Drug Molecule Representation

Table 5.2: Atom features in the drug molecule.

Feature Name	Dim	Description
Atom element	44	One-hot encoding of atom element
Degree of the atom	11	One-hot encoding of the degree of atoms
Number of Hydrogens	11	Number of hydrogen atoms
Number of implicit Hydrogens	11	Number of implicit hydrogen bonds
Aromatic	1	The atom is aromatic or not

The sequence data of the drug molecule is the SMILES string in which each character represents atoms or bonds of the molecule. To represent drug molecules as graphs, a common approach is to convert their chemical notations into graph representations. This involves mapping atoms to nodes and chemical bonds to edges. We pass the SMILES string to the RD-Kit [102] tool to generate the drug molecular graph. We initialize the nodes of the graph with the same features used in DGraphDTA [157]. The features are listed in Table 5.2, and the total feature dimension is 78.

Target Representation

Each character in the target sequence corresponds to an amino acid. To construct graph representations, we consider the structural features and interactions among amino acids. Unlike

drug molecules, protein structure generation is more complex. The current state-of-the-art, AlphaFold2 [46], predicts 3D structures from amino acid sequences.

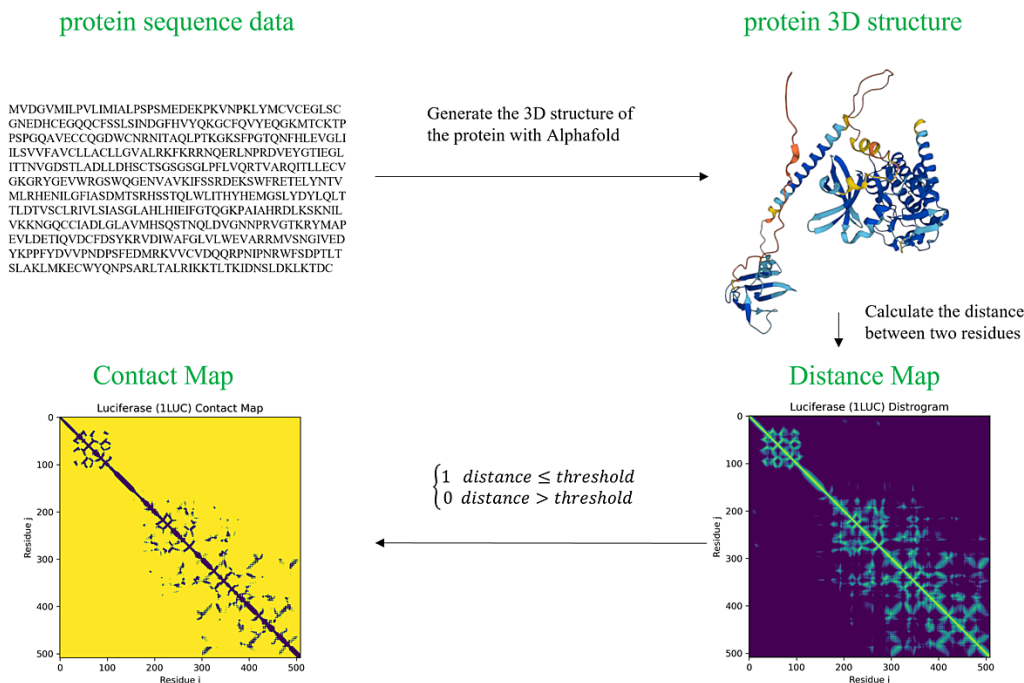


Figure 5.1: The process of the generation of protein graph with AlphaFold2.

As shown in Figure 5.1, we obtain target structures in PDB format from AlphaFold2 outputs and compute Euclidean distances between C_α atoms to build the distance map. A binary contact map is derived by applying an $8\text{\AA}$ threshold, following Duarte et al. [163]; two other thresholds are also evaluated (see Supplementary Table 1). Node features follow Jiang et al. [157], detailed in Table 5.3.

Position Probability Matrix (PPM) is a simplified variant of the Position-Specific Scoring Matrix (PSSM), commonly used to assess protein sequence similarity [164]. It is derived from multiple sequence alignment (MSA), which aligns the query sequence with thousands of homologs [165]. PPM captures the probability of each residue at each position:

$$M_{r,k} = \frac{F_{r,k} + \frac{p}{4}}{N + p}, \quad (5.2)$$

Table 5.3: Protein molecular graph feature initialization.

Feature Name	Dim	Description
Residual symbol	21	One-hot encoding of the residue
PPM	21	Position Probability Matrix
Aliphatic	1	The residue is aliphatic or not
Aromatic	1	The residue is aromatic or not
Polar neutral	1	The residue is polar neutral or not
Acidic charged	1	The residue is acidic charged or not
Basic charged	1	The residue is basic charged or not
Residue weight	1	The weight of the residue
Dissociation constant for $-\text{COOH}$ group	1	$-\log(\text{dissociation constant})$ for COOH group
Dissociation constant for $-\text{NH}_3$ group	1	$-\log(\text{dissociation constant})$ for NH_3 group
Dissociation constant for other groups	1	$-\log(\text{dissociation constant})$ for other groups
pH at isoelectric point	1	The pH at the isoelectric point
Hydrophobicity (pH=2)	1	Hydrophobicity of the residue at pH 2
Hydrophobicity (pH=7)	1	Hydrophobicity of the residue at pH 7

where $F \in \mathbb{N}^{R \times K}$ is the residue frequency matrix for R residue types and sequence length K . $F_{r,k}$ is the frequency of residue r at position k , N is the total number of aligned sequences, and p is a pseudocount.

5.3 Methods

Figure 5.2 illustrates the proposed model architecture, which can be divided into three main components: Input, Encoding, and Prediction. In the input part, graph data is generated along with the input sequence data. The encoding part comprises three models. The Chemformer [149] and ESM [74] models are pre-trained for feature extraction from drug and protein sequence data, respectively. Additionally, the GNN component utilizes a k -subgraph GNN extractor as proposed in NGNN [160], which generates k -hop subgraphs such that each node encodes a local subgraph.

This hierarchical representation enhances the expressiveness by first building subgraph-level representations and then assembling them into the overall graph. To capture interaction information between drugs and targets, we employ a cross-attention-free transformer (cross-

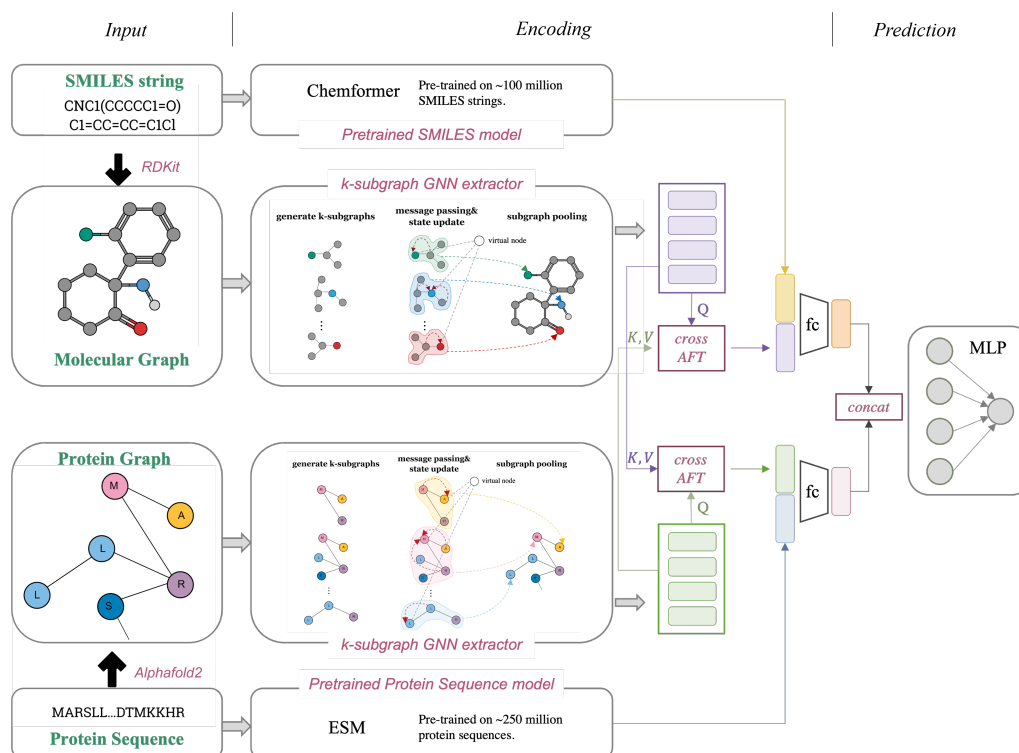


Figure 5.2: **Model architecture of iNGNN-DTI.** The SMILES string and the protein amino acid sequence are transformed into the graph data in the Input section. The Encoding section uses the k -subgraph GNN extractor to encode the graph data while using the Chemformer and ESM models to encode the SMILES string and amino acid sequence, respectively. To learn the interaction information, the cross-AFT module connects to the GNN extractor. The final representation of the drug and the protein is created by concatenating the output of the cross-AFT module with the global feature created by the unsupervised model. In the end, the two representations are concatenated and passed to a MLP module for the prediction of the interaction.

AFT) [161] on top of the GNN output. In the prediction part, all extracted features are integrated and passed into a multilayer perceptron (MLP) to make predictions. The complete architecture is referred to as **iNGNN-DTI**. The following subsections describe each module in detail.

5.3.1 k -subgraph GNN extractor layer with virtual node

To encode the graph data, we utilize the k -subgraph GNN extractor, as introduced in NGNN [160]. Traditional GNN models such as GCN exhibit limitations in expressing substructures when the

graph is nested rather than a tree structure [166]. The k -subgraph GNN extractor addresses this limitation by updating the representation of each node using the k -hop subgraph centered at that node, resulting in more powerful representations compared to base GNNs.

The implementation of the k -subgraph GNN extractor involves incorporating a subgraph pooling layer on top of the base GNN layers. This pooling operation aggregates information from the k -hop subgraph and propagates it to update the node representations.

We denote the graph data as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges. An intermediate node representation is first generated using a base layer based on GCN. The intermediate representation of node u at layer $l + 1$ is calculated as:

$$h_u^{(l+1)} = \sigma \left(\sum_{j \in \mathcal{N}(u)} W^{(l)} h_j^{(l)} \right), \quad (5.3)$$

where $h_u^{(l+1)} \in \mathbb{R}^d$ denotes the representation of node u , and d is the feature dimension. $\sigma(\cdot)$ is a non-linear activation function. $\mathcal{N}(u)$ denotes the set of neighboring nodes of u , and $W^{(l)} \in \mathbb{R}^{d \times d}$ is the learnable weight matrix at layer l .

The final representation is obtained by applying a subgraph pooling operation for each node following the base GNN layers. The pooling is defined as:

$$h_u = \sum_{k \in \mathcal{S}(u)} h_k, \quad (5.4)$$

where $\mathcal{S}(u)$ denotes the set of nodes in the k -hop subgraph centered at node u .

A virtual node is an additional node inserted into the graph. Virtual edges are created to link the virtual node to all other nodes in the graph [48]. After each message-passing layer, the virtual node aggregates the information of all nodes and scatters it back to them before the next update. The update of the virtual node and its interaction with other nodes are computed as follows:

$$v^{(l)} = \sigma \left(v^{(l-1)} W_0^{(l)} + \left(\sum_{i \in \mathcal{V}} h_i^{(l)} \right) W_1^{(l)} \right), \quad (5.5)$$

$$h_u^{(l+1)} = h_u^{(l+1)} + v^{(l)}, \quad (5.6)$$

where $v^{(l)}$ is the representation of the virtual node at layer l , and $W_0^{(l)}, W_1^{(l)} \in \mathbb{R}^{d \times d}$ are learnable weight matrices.

5.3.2 Interaction module

The interaction module is designed to capture the molecular interactions between drugs and their target proteins, highlighting key atoms in the drug molecule and showcasing the attention patterns in the target during the interaction. This module is implemented using an attention-free transformer (AFT) [161]. Unlike classical transformers [120], AFT introduces a transformer operation that removes the scaled dot-product attention mechanism. Given an input X , the AFT operates as:

$$Y_t = \sigma_q(Q_t) \odot \frac{\sum_{t'=1}^T \exp(K_{t'} + w_{t,t'}) \odot V_{t'}}{\sum_{t'=1}^T \exp(K_{t'} + w_{t,t'})}, \quad (5.7)$$

where Q , K , and V are the query, key, and value matrices derived from linear projections: $Q = XW^Q$, $K = XW^K$, and $V = XW^V$. $w \in \mathbb{R}^{T \times T}$ is the pairwise positional bias matrix learned during training. σ_q is the sigmoid activation function, $\odot$ denotes element-wise multiplication, and t indexes the position. In our implementation, we extend this to a ****cross-AFT**** setting.

Let the outputs of the GNN module for the drug and protein branches be $H_{\text{drug}} \in \mathbb{R}^{N_{\text{drug}} \times d}$ and $H_{\text{prot}} \in \mathbb{R}^{N_{\text{prot}} \times d}$, respectively, where N_{drug} and N_{prot} denote the number of nodes in the drug and protein graphs. The cross-AFT applied on the drug branch is computed as follows:

$$Q_{\text{drug}} = H_{\text{drug}} W_{\text{drug}}^Q, \quad K_{\text{prot}} = H_{\text{prot}} W_{\text{prot}}^K, \quad V_{\text{prot}} = H_{\text{prot}} W_{\text{prot}}^V, \quad (5.8)$$

$$Y_{\text{drug}}^n = \sigma_q(Q_{\text{drug}}^n) \odot \frac{\sum_{n'=1}^{N_{\text{prot}}} \exp(K_{\text{prot}}^{n'} + w_{n,n'}) \odot V_{\text{prot}}^{n'}}{\sum_{n'=1}^{N_{\text{prot}}} \exp(K_{\text{prot}}^{n'} + w_{n,n'})}, \quad (5.9)$$

$$Y_{\text{drug}} = Y_{\text{drug}} + H_{\text{drug}}, \quad (5.10)$$

where $w \in \mathbb{R}^{N_{\text{drug}} \times N_{\text{prot}}}$ denotes the pairwise bias matrix between drug and protein nodes, and n indexes nodes in the drug graph. The cross-AFT process on the protein branch follows the same procedure, with the roles of drug and protein features swapped.

5.3.3 Merge features extracted from pre-trained models

Molecular features extracted by pre-trained models may encode fundamental chemical or biological properties. In this work, we employ Chemformer [149] for drug molecule feature extraction and ESM [74] for protein sequences. Both models take sequence data as input.

Chemformer is a BART-based model [167] pre-trained on over 100 million SMILES strings. It produces a 512-dimensional embedding for each molecule. ESM is a 34-layer transformer model trained on 250 million amino acid sequences, generating a 768-dimensional embedding per protein. To align dimensions, we map both outputs to a 128-dimensional space using fully connected (FC) layers.

Let P_{drug} and P_{prot} denote the extracted features from SMILES and protein sequences, respectively. These features are fused with the output of the interaction module using concatenation followed by an FC layer to obtain the final representations:

$$F_{\text{drug}} = \text{FC}([Y_{\text{drug}} \cdot P_{\text{drug}}]), \quad (5.11)$$

$$F_{\text{prot}} = \text{FC}([Y_{\text{prot}} \cdot P_{\text{prot}}]), \quad (5.12)$$

where $[\cdot]$ denotes the concatenation operation.

Finally, the drug and protein representations are concatenated and passed through a multi-layer perceptron (MLP) to predict the interaction:

$$\text{output} = \text{MLP}\left([F_{\text{drug}} \cdot F_{\text{prot}}]\right). \quad (5.13)$$

5.4 Baseline Models and Model Evaluation

To evaluate the performance of our proposed model, we compare it with four baseline models: DeepDTA [37], ML-DTI [155], MolTrans [156], and DGraphDTA [157]. DeepDTA employs two 3-layer convolutional neural networks (CNNs) for processing drug SMILES and protein sequences. MolTrans adopts a transformer-based architecture with an interaction matrix following feature extraction. ML-DTI introduces an interaction module between CNN layers but focuses its attention computation only on the drug molecule. DGraphDTA utilizes graph neural networks (GNNs) for both drug and protein representations. The original implementation of DGraphDTA leverages PconsC4 [168] to construct protein structures; however, in our experiments, we substitute this with AlphaFold2-generated structures for consistency.

All baseline models are evaluated using the recommended hyperparameter settings from their respective original studies. We use four standard performance metrics for drug-target interaction prediction: area under the receiver operating characteristic curve (AUROC), area under the precision-recall curve (AUPRC), sensitivity, and specificity. Each dataset is randomly split into training, validation, and testing sets in a 7:1:2 ratio. For each experiment, we perform five independent runs and report the mean and standard deviation of the results.

5.5 Results

5.5.1 Performance for DTI prediction

As shown in Table 5.4, our method achieves the highest performance according to AUROC and AUPRC among all methods on all three datasets. For the **Davis** dataset, we observe improvements in AUROC of 2.3% (from 0.910 to 0.931) and AUPRC of 23.8% (from 0.382 to 0.473).

Table 5.4: Model performance comparison.

Method	AUROC	AUPRC	Sensitivity	Specificity
DAVIS dataset				
DeepDTA	0.892 ± 0.0066	0.378 ± 0.0231	0.854 ± 0.0066	0.792 ± 0.0291
Moltrans	0.898 ± 0.0050	0.371 ± 0.0067	0.865 ± 0.0050	0.783 ± 0.0387
ML-DTI	0.910 ± 0.0034	0.381 ± 0.0247	0.895 ± 0.0034	0.795 ± 0.0183
DGraptDTA (AlphaFold2)	0.885 ± 0.0099	0.316 ± 0.0447	0.894 ± 0.0099	0.724 ± 0.0467
iNGNN-DTI	0.931 ± 0.0027	0.473 ± 0.0167	0.922 ± 0.0155	0.802 ± 0.0240
KIBA dataset				
DeepDTA	0.912 ± 0.0037	0.743 ± 0.0127	0.881 ± 0.0056	0.780 ± 0.0127
Moltrans	0.899 ± 0.0026	0.691 ± 0.0142	0.872 ± 0.0116	0.760 ± 0.0160
ML-DTI	0.909 ± 0.0020	0.727 ± 0.0108	0.878 ± 0.0111	0.779 ± 0.0113
DGraptDTA (AlphaFold2)	0.911 ± 0.0044	0.739 ± 0.0043	0.881 ± 0.0183	0.784 ± 0.0277
iNGNN-DTI	0.915 ± 0.0016	0.753 ± 0.0071	0.888 ± 0.0107	0.779 ± 0.0146
BIOSNAP dataset				
DeepDTA	0.897 ± 0.0274	0.900 ± 0.0046	0.859 ± 0.0089	0.786 ± 0.0197
Moltrans	0.887 ± 0.0354	0.881 ± 0.0085	0.824 ± 0.0106	0.809 ± 0.0104
ML-DTI	0.911 ± 0.0053	0.911 ± 0.0126	0.851 ± 0.0054	0.828 ± 0.0215
DGraptDTA (AlphaFold2)	0.918 ± 0.0052	0.917 ± 0.0024	0.858 ± 0.0175	0.831 ± 0.0151
iNGNN-DTI	0.934 ± 0.0021	0.939 ± 0.0022	0.872 ± 0.0189	0.854 ± 0.0200

For the **BIOSNAP** dataset, the improvements are 2.3% (from 0.913 to 0.934) and 2.4% (from 0.917 to 0.939) for AUROC and AUPRC, respectively. The improvements on the **KIBA** dataset are 0.3% (from 0.912 to 0.915) for AUROC and 1.3% (from 0.743 to 0.753) for AUPRC.

5.5.2 Performance on the Test Set with Unseen Drugs and Targets

In practice, it is common for new drugs or targets to be discovered while their interactions have not yet been determined. To ensure reliable DTI predictions in such cases, unseen input data is extracted from the DAVIS dataset. We follow the experimental setup in MolTrans [156], where 20% of drugs or targets are randomly selected and all related DTI samples are used as the test set. The remaining samples are divided into training and validation sets with a 7:1 ratio. All

Table 5.5: Model performance on the test set with unseen drugs and proteins.

Method	Unseen Drugs		Unseen Proteins	
	AUROC	AUPRC	AUROC	AUPRC
DeepDTA	0.736 $\pm$ 0.0550	0.145 $\pm$ 0.0300	0.770 $\pm$ 0.0594	0.145 $\pm$ 0.0300
Moltrans	0.744 $\pm$ 0.0540	0.144 $\pm$ 0.0370	0.778 $\pm$ 0.0538	0.231 $\pm$ 0.0534
ML-DTI	0.737 $\pm$ 0.0700	0.169 $\pm$ 0.0730	0.840 $\pm$ 0.0357	0.259 $\pm$ 0.0519
DGraphDTA	0.718 $\pm$ 0.0045	0.169 $\pm$ 0.0049	0.780 $\pm$ 0.0478	0.164 $\pm$ 0.0391
iNGNN-DTI	0.763 $\pm$ 0.0490	0.224 $\pm$ 0.0640	0.867 $\pm$ 0.0357	0.296 $\pm$ 0.0534

baseline methods are evaluated on the unseen datasets.

The results are summarized in Table 5.5. When the test set does not include any unseen samples, ML-DTI achieves the best performance among the four baseline methods on the DAVIS dataset (Table 5.4). However, in the presence of unseen drugs, ML-DTI performs comparably to DeepDTA and MolTrans. In contrast, our proposed method demonstrates more consistent performance across different scenarios. Specifically, for the unseen drug case, we achieve improvements in AUROC of 2.5% (from 0.744 to 0.763) and AUPRC of 32.5% (from 0.169 to 0.224). Similarly, for the unseen protein case, we observe improvements in AUROC of 3.1% (from 0.840 to 0.867) and AUPRC of 14.3% (from 0.259 to 0.296).

5.5.3 Interpretability Analysis

To enhance the interpretability of the prediction results, we visualize the information learned by the interaction module, specifically the AFT module. Unlike explicit attention matrices, the AFT module does not directly learn an attention matrix. However, it can be interpreted as having an implicit attention mechanism through its operations. The interpretation of the AFT module with implicit attention can be understood as follows:

$$Y_{drug}^{n,i} = \langle a_{drug}^{n,i}, V_{prot}^i \rangle, \quad (5.14)$$

$$\text{s.t. } a_{drug}^{n,i} = \frac{\sigma_q(Q_{drug}^{n,i}) \exp(K_{prot}^{n,i} + w_n)}{\sum_{n'=1}^{N_{prot}} \exp(K_{prot}^{n',i} + w_{n,n'})}, \quad i = 1, 2, \dots, d, \quad n = 1, 2, \dots, N_{prot}. \quad (5.15)$$

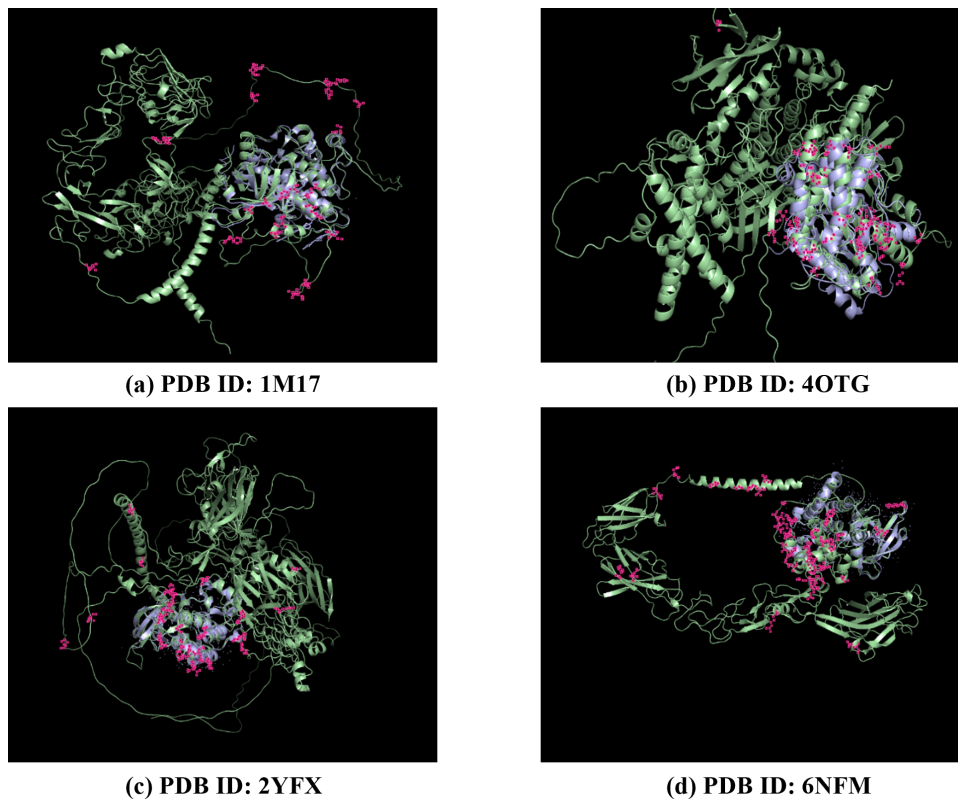


Figure 5.3: **Visualization of the residues with the highest weight in each dimension on the target structure.** The complete structure of the protein predicted by AlphaFold2 is denoted by the green structures, while the purple structures represent the structure sourced from the Protein Data Bank (PDB). Residues with the highest scores in each dimension are marked by pink dots.

Here, i indexes the feature dimension, and $a_{drug}^{n,i} \in \mathbb{R}^{N_{prot}}$ represents the attention vector for each dimension. Two attention matrices, $A_{drug} \in \mathbb{R}^{N_{drug} \times N_{prot} \times d}$ and $A_{prot} \in \mathbb{R}^{N_{prot} \times N_{drug} \times d}$, are generated by using drugs and targets as queries, respectively. The following subsections analyze these matrices via visualization and virtual docking.

Visualization Analysis of the Targets

We randomly select four protein targets with both predicted and reference structures. Residues with the highest attention scores in each dimension are shown as pink dots in Figure 5.3. The green structure denotes the full-length protein from AlphaFold2 (used as model input), while the purple segments represent the reference PDB complex. The full structure is typically much

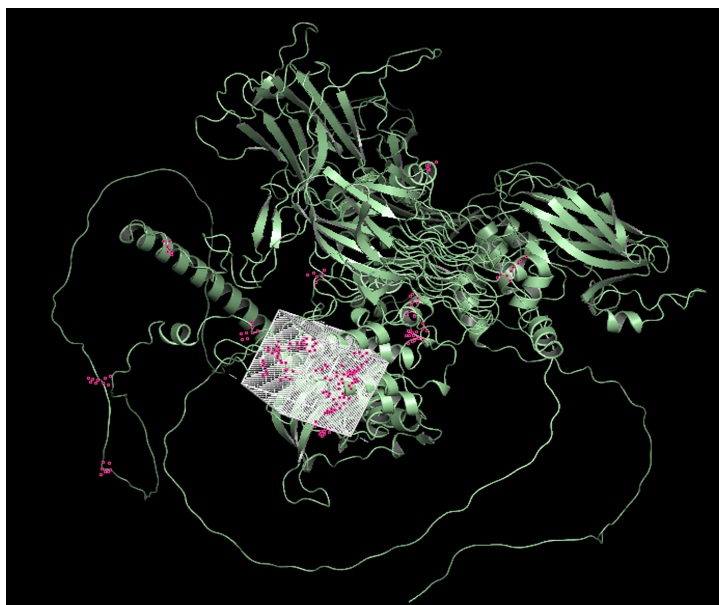


Figure 5.4: **Grid box configuration for virtual docking on 2YFX.**

longer than the complex, making binding site identification challenging—especially without explicit site annotations. Some high-weight residues appear in loop regions, likely due to their prevalence and structural similarity, but a clear clustering around interaction regions is observed. This is particularly evident in 4OTG (Figure 5.3(b)), where most residues concentrate near the binding interface.

Virtual Docking Analysis

We further perform virtual docking tests by placing the grid box across the area where most residues with higher scores are clustered. Figure 5.4 displays the grid box utilized for the structure 2YFX case. The virtual docking results based on AutoDock Vina [18] are shown in Figure 5.5. The ligand in blue (the true target structure in purple) is sourced from the PDB, while the green ligand is derived from our virtual docking result. Both cases achieved moderate virtual docking scores (-8.9 kcal/mol and -7.2 kcal/mol), and for the 2YFX case, our virtual docking position closely aligns with the position provided by the PDB.

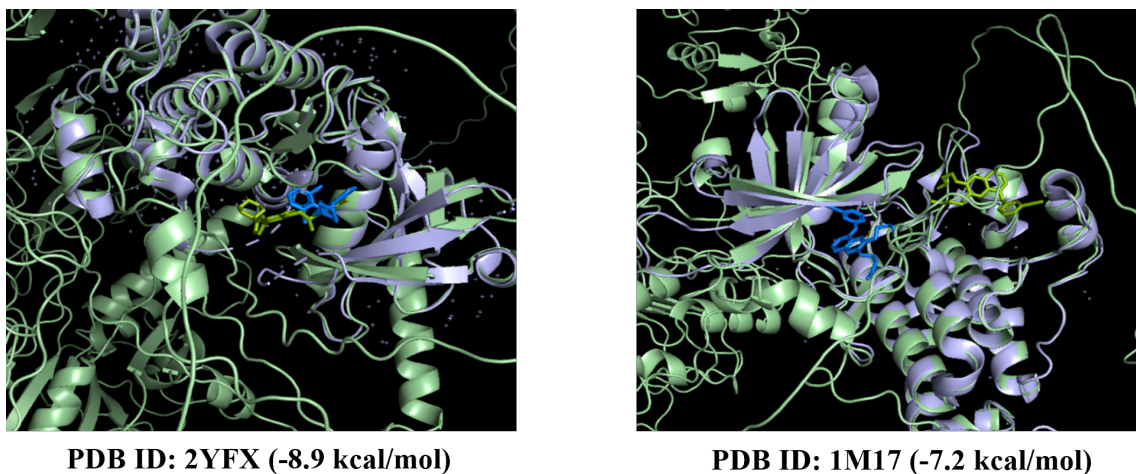


Figure 5.5: **Visualization of the virtual docking result.** The ligand in blue is sourced from the PDB, while the green ligand is derived from our virtual docking result.

Visualization Analysis on the Drug Molecules

We visualize atoms with the highest weight on the drug molecule (Figure 5.6). In the 1M17 complex, the target protein forms hydrogen bonds with N2 and N3 in the drug molecule, both highlighted by our attention mechanism. Similarly, in 4OTG, O2 and N3 engage in hydrogen bonds with the target protein, with one of them emphasized by our attention mechanism.

5.5.4 Ablation Study

Effective of different modules

Table 5.6: Ablation test on the DAVIS dataset.

No. of tests	GNN	NGNN	cross AFT	unsupervised model	AUROC	AUPRC
1	✓				0.916 ± 0.0046	0.447 ± 0.0181
2		✓			0.919 ± 0.0037	0.464 ± 0.0357
3		✓	✓		0.920 ± 0.0051	0.478 ± 0.0222
4		✓		✓	0.927 ± 0.0052	0.431 ± 0.0103
5		✓	✓	✓	0.931 ± 0.0027	0.473 ± 0.0167

There are two modifications on top of the basic GNN networks: the interaction module using the cross-AFT mechanism and the use of features extracted by an unsupervised model. We conduct an ablation study to assess the effectiveness of each component using the Davis dataset.

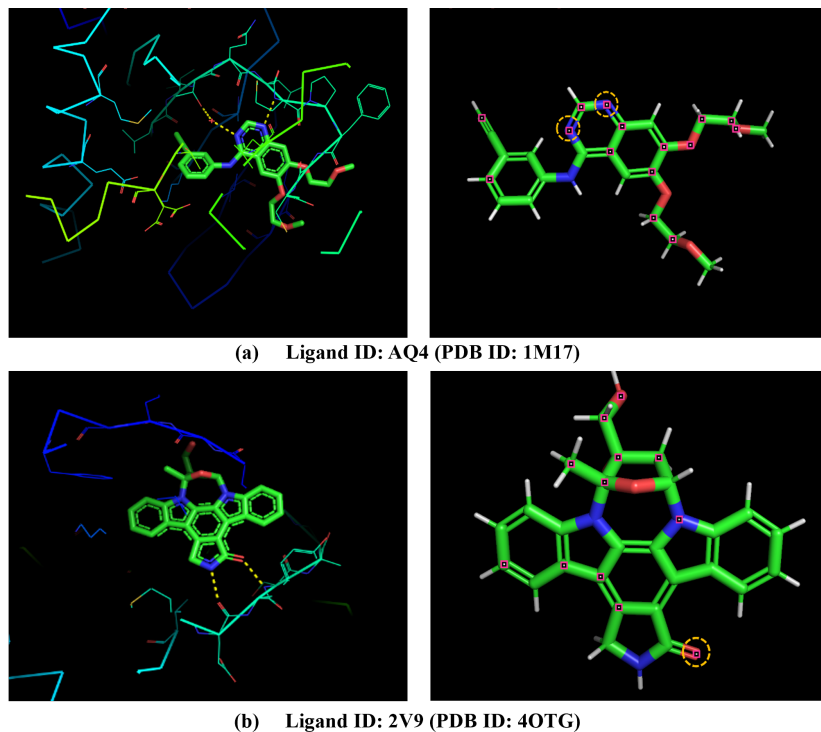


Figure 5.6: **Visualization of the atoms with the highest weight in each dimension on the drug molecule.** In each panel, known polar contacts are illustrated with yellow dashed lines on the left side figure. On the right side, atoms with high scores are denoted by pink points, while atoms forming bonds with the target are enclosed within yellow circles.

As shown in Table 5.6, we observe that the cross-AFT module enhances the AUPRC result, and the incorporation of features learned by unsupervised models contributes to improvements in AUROC. Combining both modules leads to further gains compared to the baseline NGNN. Specifically, AUROC increases by 1.3% (from 0.919 to 0.931), and AUPRC increases by 1.9% (from 0.464 to 0.473).

Performance Comparison of Two Different Contact Map Generation Methods

With AlphaFold2 showcasing impressive accuracy in predicting protein structures, it is anticipated that the resulting contact maps would exhibit a high level of quality. We experiment with SPOT-Contact-LM [169] to predict the contact maps and replace those generated by AlphaFold2, using the DAVIS dataset.

However, as shown in Figure 5.7, this configuration results in an AUROC (Area Under

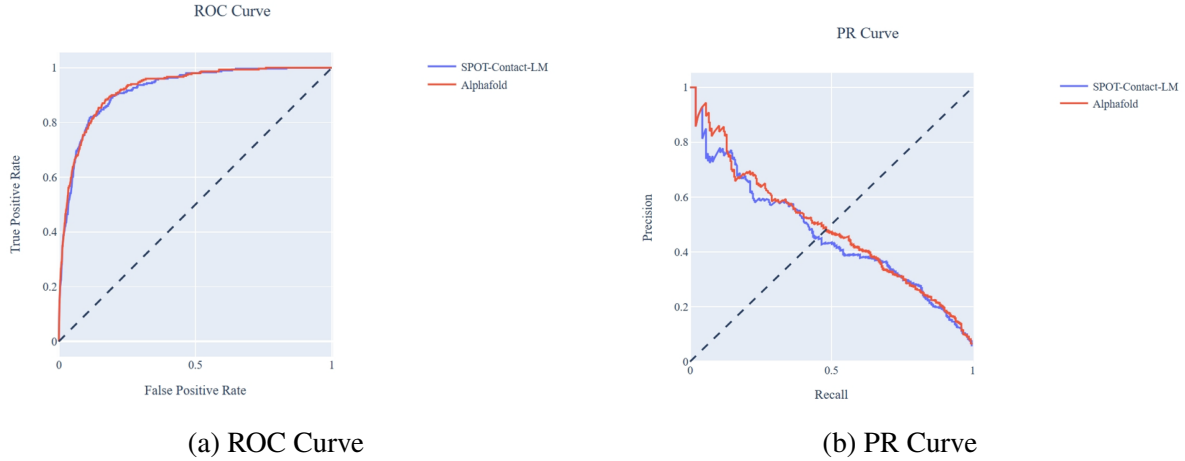


Figure 5.7: ROC and PR curves comparing two contact map generation methods on the DAVIS dataset: AlphaFold2 vs. SPOT-Contact-LM.

the ROC Curve) of 0.924 ± 0.0064 (mean $\pm$ standard deviation) and an AUPRC (Area Under the Precision-Recall Curve) of 0.458 ± 0.0127 , which is slightly worse than using contact maps directly generated by AlphaFold2 (0.931 ± 0.0027 for AUROC and 0.473 ± 0.0167 for AUPRC).

Comparison of performance between the membrane and non-membrane proteins across three datasets

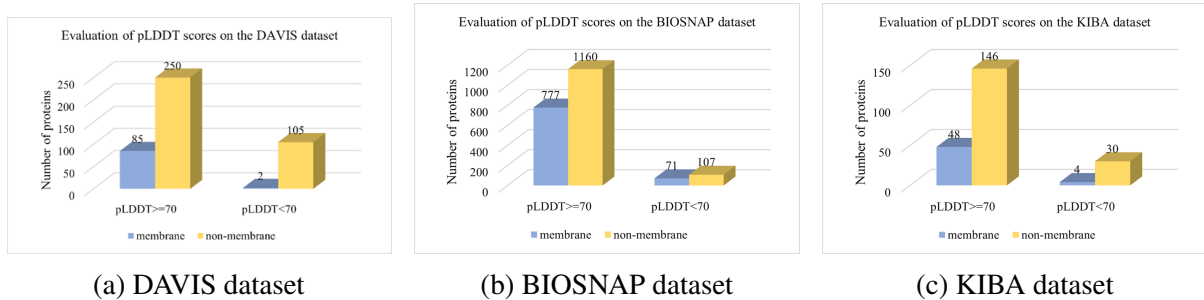


Figure 5.8: Comparison of pLDDT scores between membrane and non-membrane proteins across three datasets. Proteins are classified as high-confidence ($pLDDT > 70$) or low-confidence ($pLDDT \leq 70$).

The evaluation results presented by Hegedűs et al. (2022) [170] indicate that AlphaFold2 exhibits comparable performance for membrane proteins as it does for soluble proteins. In our evaluation of membrane proteins within our datasets, we consider the predicted Local Distance

Difference Test (pLDDT) score. A pLDDT score of 90 or higher is generally considered to indicate very high quality, while scores in the range of 70 to 90 are considered confident in structure predictions. Scores below 70 are indicative of low confidence [46].

Specifically, the DAVIS dataset comprises 87 membrane proteins and 355 non-membrane proteins. The BIOSNAP dataset includes 848 membrane proteins and 1267 non-membrane proteins, while the KIBA dataset contains 52 membrane proteins and 176 non-membrane proteins. For each protein, we compute the mean pLDDT score and classify the proteins based on whether their scores are greater than 70 or less than 70, as depicted in Figure 5.8.

In the DAVIS dataset, 97.7% of membrane proteins received scores above 70, while 70.4% of non-membrane proteins also scored above 70. For the BIOSNAP dataset, the corresponding percentages are 91.6% and 91.6%, respectively. In the KIBA dataset, the values are 92.3% and 83.0%, respectively. Using Fisher’s exact test, we did not observe a significant difference in the distribution of high and low confidence predictions between membrane and non-membrane proteins for both BIOSNAP (p-value = 1.00) and KIBA (p-value = 0.12). However, a significant difference was observed in the DAVIS dataset (p-value < 0.01).

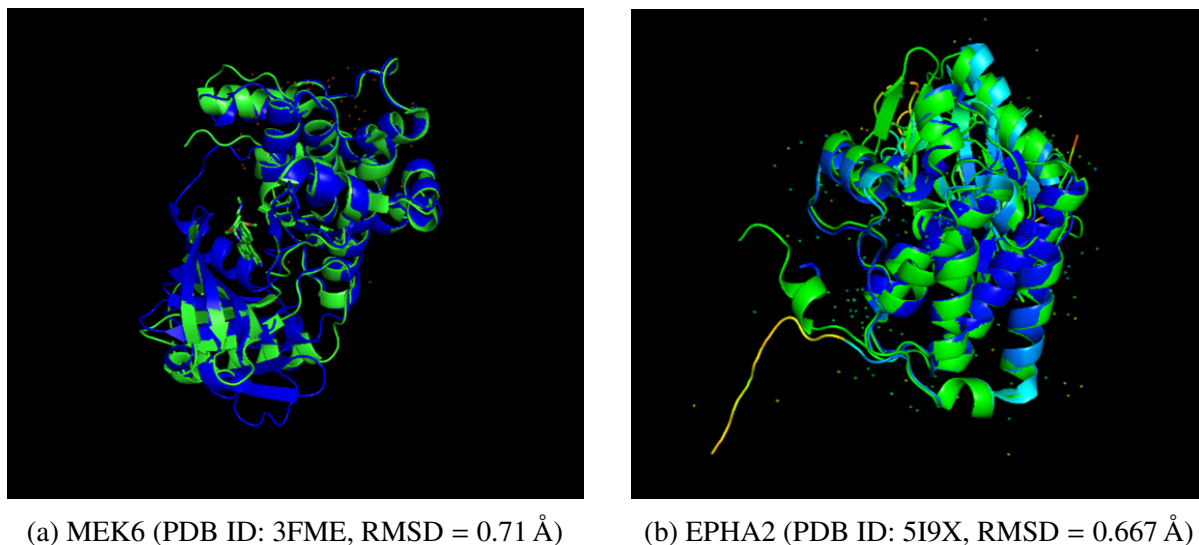


Figure 5.9: Visualization of the alignment between AlphaFold-predicted structures (blue) and corresponding PDB experimental structures (green).

Furthermore, we conducted a structural comparison of two membrane proteins—MEK6

and EPHA2—by aligning AlphaFold2-predicted models to their experimentally resolved structures using PyMOL. The results are presented in Figure 5.9. The root mean square deviation (RMSD) values for the two examples were 0.71 Å and 0.667 Å, respectively. RMSD serves as a metric for assessing structural similarity by calculating the average distance between equivalent atoms in two aligned structures. Typically, an RMSD below 2 Å is considered indicative of good structural agreement [171, 172].

Performance Comparison of the DAVIS Dataset with Varied Cut-Offs for Binarization of the Contact Map

Table 5.7: Performance on the DAVIS dataset with varied cut-offs for binarization of the contact map.

Threshold	6 Å	8 Å	10 Å
AUROC	0.928 ± 0.0045	0.931 ± 0.0027	0.923 ± 0.0063
AUPRC	0.472 ± 0.0142	0.473 ± 0.0167	0.447 ± 0.0244

Duarte et al. (2010) explored the impact of the cut-off threshold on contact maps for protein structure reconstruction [163]. The findings indicate that beyond 8 Å, there is no significant improvement in the accuracy of the reconstructed protein structure, suggesting that larger thresholds may not contribute additional structural information.

To validate this in our task, we conduct experiments with two additional cut-offs—specifically 6 Å and 10 Å—on the DAVIS dataset. As shown in Table 5.7, both the 6 Å and 10 Å thresholds result in slightly lower performance compared to the default 8 Å threshold.

5.5.5 COVID-19 Case Study

Coronavirus Disease 2019 (COVID-19) is caused by Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2) [173]. Numerous proteins have been implicated in the pathogenesis of COVID-19. In this case study, we explore the potential of repurposing drugs that interact with angiotensin-converting enzyme 2 (ACE2), a receptor protein expressed on the

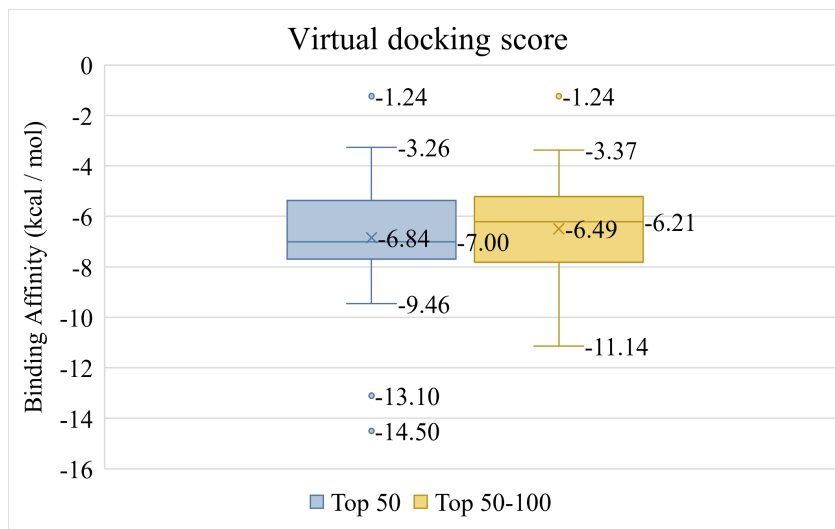


Figure 5.10: **Virtual docking scores for top 50 drugs, and top 50-100 drugs.**

surface of human cells [174]. This protein serves as the entry point for SARS-CoV-2 into host cells.

We pre-train our model using the BIOSNAP dataset and predict interactions between ACE2 and all 4,510 drugs within BIOSNAP. Due to the limited availability of experimental validations, we perform blind virtual docking to evaluate potential drug-ACE2 interactions using DockConv2 [175], which scans the whole protein structure to identify possible binding sites.

Additionally, we perform grid-based virtual docking focusing on residue clusters with higher attention scores, as described previously. Figure 5.10 illustrates the blind virtual docking scores for the top 50 drugs and those ranked between 50 and 100. The mean (median) docking score for the top 50 drugs is -6.84 (-7.00) kcal/mol, whereas for the 50–100 ranked drugs, it is -6.49 (-6.21) kcal/mol, suggesting better binding affinities among the top-ranked predictions.

Among the top five drugs identified by our model, Pseudoephedrine and Ephedrine have been reported as inhibitors of the ACE2–SARS-CoV-2 spike protein interaction [176, 177]. Yamamoto et al. [178] identified Marimastat, a metalloproteinase inhibitor, as a promising COVID-19 therapeutic. Additionally, erythromycin achieved the strongest binding affinity (-14.5 kcal/mol) and has been recognized as a high-potential COVID-19 treatment [179].

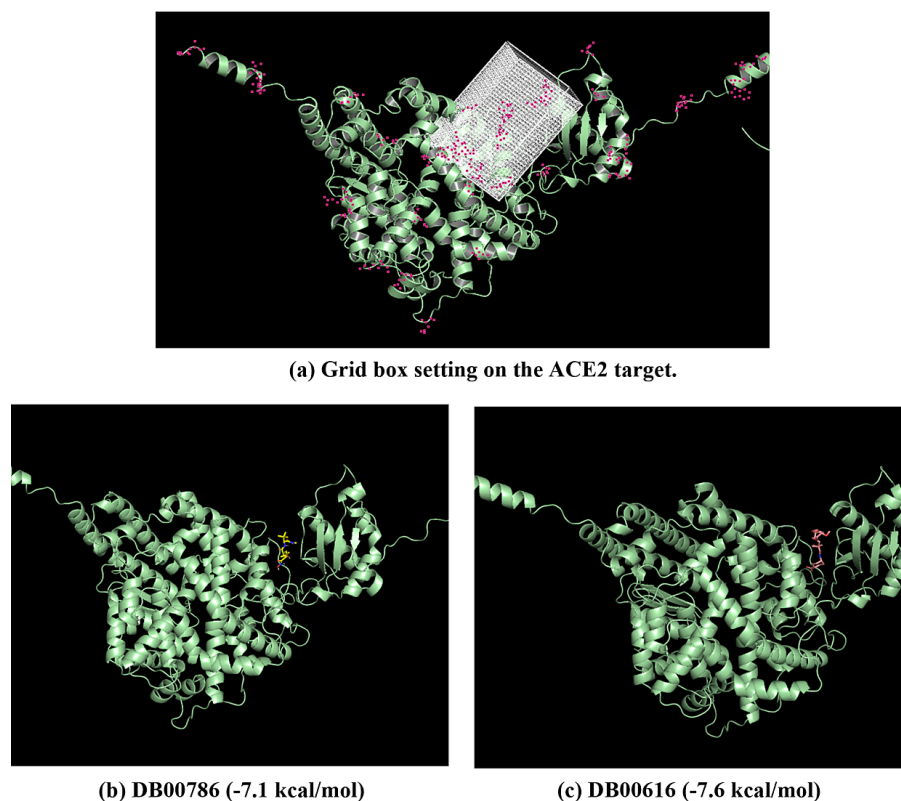


Figure 5.11: Visualization of grid box-based virtual docking results for drugs DB00786 and DB00616 on the ACE2 target.

Figure 5.11 shows the grid box setting on ACE2. We perform docking for two top-ranked drugs (DB00786 and DB00616), and their docking results are shown in Figures 5.11(a) and 5.11(b). The resulting docking scores are -7.1 and -7.6 kcal/mol, respectively. Notably, the docking positions closely match those generated by DockConv2 blind docking (Figure 5.12).

5.6 Summary

Our work introduced a novel interpretable nested graph neural network for the prediction of drug–target interactions. We utilized AlphaFold2 to generate molecular structures of target proteins and constructed graphs based on these structures. The model integrated features extracted

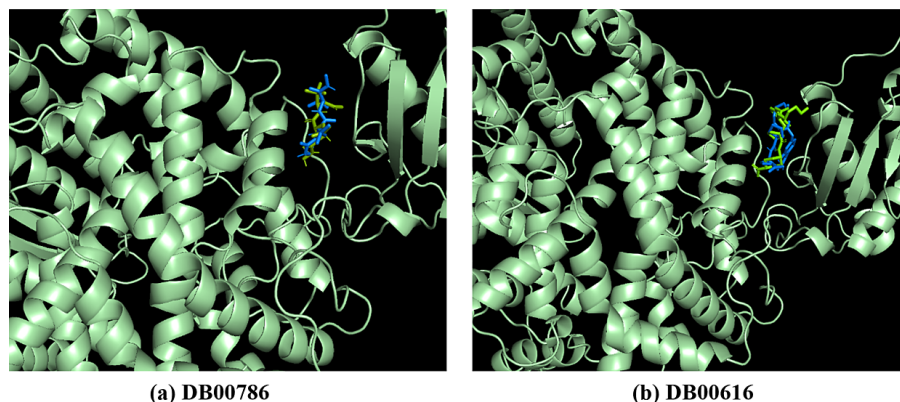


Figure 5.12: **Comparison of blind docking results and our grid box-based virtual docking results.** The blue ligand represents the outcome from blind docking, while the green ligand represents the results from our grid box-based virtual docking analysis.

from 1D sequence data via pre-trained models and features learned from 2D molecular graphs using a graph neural network. To capture drug–target interaction information, we incorporated an attention-free transformer module.

Our proposed architecture achieved superior performance compared to four baseline models across all test datasets. In a case study focused on COVID-19, we repurposed five drugs, three of which had previously been identified as promising candidates for COVID-19 treatment. This demonstrated the potential of our approach in drug discovery tasks.

From the interpretability analysis of our proposed model, we observed that the model could highlight certain atoms within some drugs that participated in hydrogen bonds or water-mediated bonds. However, we also noted that the attention weights assigned to the protein target did not always exhibit a strong correlation with the actual positions of the residues, which indicated the need for further improvement in our future work.

Future work might focus on two directions. First, further analysis could evaluate our algorithm’s performance on both membrane and non-membrane proteins using AlphaFold2-predicted structures, as preliminary results suggested high accuracy for most membrane pro-

teins (Supplementary Figures [5.8](#) and [5.9](#)). Second, since generating contact maps with AlphaFold2 was computationally expensive, alternative sequence-based methods could be explored.

Chapter 6

Case Study: Discovery of PIN1 Inhibitors

6.1 Introduction

Triple-negative breast cancer (TNBC) is a highly aggressive subtype of breast cancer characterized by the absence of estrogen receptor (ER), progesterone receptor (PR), and HER2 expression. Although TNBC represents only about 10% of all breast cancers, it is associated with poor prognosis, high relapse rates, and limited therapeutic options [180]. Recent advances in cancer immunotherapy, such as immune checkpoint inhibitors (ICIs) and antibody-drug conjugates, have improved treatment outcomes for several cancer types [181, 182], yet many patients, including those with TNBC, derive limited benefit due to intrinsic resistance mechanisms [183, 184].

A major challenge in TNBC is the enrichment of cancer stem-like cells (CSCs) and an immunosuppressive tumor microenvironment (TME), both of which contribute to therapy resistance and recurrence [185, 186]. A key molecular driver in this context is the prolyl isomerase PIN1, which specifically recognizes and isomerizes phosphorylated Ser/Thr-Pro motifs (Figure 6.1). This post-phosphorylation regulatory mechanism modulates the activity, stability, and interactions of a wide range of oncogenic and tumor-suppressive proteins [187, 188].

PIN1 is frequently overexpressed in breast cancer and regulates over 70 oncoproteins and

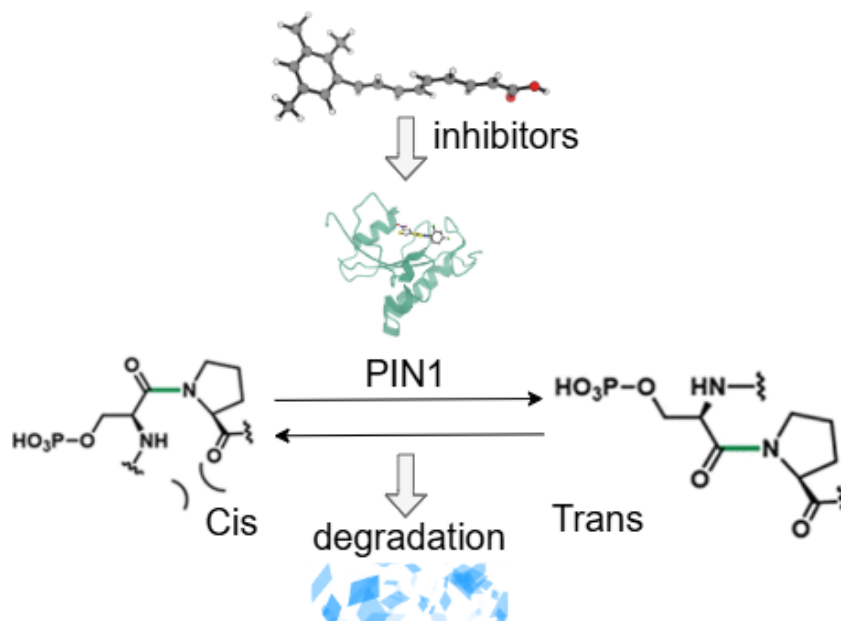


Figure 6.1: Mechanism of Pin1-mediated cis–trans isomerization and its inhibition, adapted from [1]. Pin1 catalyzes the conversion between cis and trans conformations of phosphorylated Ser/Thr-Pro motifs. Inhibitors block this activity, leading to altered protein degradation pathways.

more than 35 tumor suppressors [187, 189, 190]. It promotes tumorigenesis by sustaining proliferative signaling, inhibiting apoptosis, maintaining stemness, and reshaping the TME. In TNBC, PIN1 drives CSC expansion and supports immune evasion, highlighting its role as a central node in cancer progression [191, 192, 193].

Although several PIN1 inhibitors and substrate mimetics have been identified, clinical translation remains limited due to issues with potency, specificity, and pharmacokinetics [194, 195, 196]. The synergistic effects of all-trans retinoic acid (ATRA) and arsenic trioxide (ATO) in degrading PIN1 and eliminating CSCs in TNBC and acute promyelocytic leukemia (APL) underscore the therapeutic potential of PIN1 inhibition [190, 197], although these agents lack specificity and raise safety concerns in solid tumors.

Taken together, these findings establish PIN1 as a promising yet underexploited therapeutic target in TNBC. Targeting PIN1 could simultaneously disrupt multiple oncogenic pathways and reprogram the immunosuppressive TME. However, there remains a critical need for the development of novel, potent, and selective PIN1 inhibitors with drug-like properties suitable

for clinical application. This motivates the integration of computational approaches such as AI-driven molecular modeling to accelerate the discovery of effective PIN1-targeted therapies for TNBC.

To facilitate novel PIN1 inhibitor discovery in TNBC, we applied a virtual screening strategy integrating advanced molecular representation learning with target-specific activity prediction. In particular, we developed a hybrid model combining MolGraph-xLSTM (Chapter 3) and DistRouting (Chapter 4). MolGraph-xLSTM captures hierarchical molecular representations by integrating atom-level and motif-level graphs for multiscale feature extraction. DistRouting addresses target imbalance in molecular property prediction by guiding expert selection based on target distribution similarity. We use MolGraph-xLSTM as the base encoder to generate unified molecular representations, which are then fed into a mixture-of-experts module guided by DistRouting. This routing mechanism assigns each input to the most relevant expert sub-networks based on label distribution, allowing specialization on different property ranges (e.g., low vs. high values).

We utilized a curated dataset of PIN1 inhibitors, with potency and efficacy values derived from fluorescence polarization-based high-throughput screening assays [198]. Instead of raw potency values, we defined a composite metric—the square root of the potency-to-efficacy ratio (PE)—to balance binding strength and functional response. This continuous value served as the regression target during training to reflect both biochemical binding and cellular performance.

The combined MolGraph-xLSTM + DistRouting model was trained on this labeled dataset and then applied to screen a virtual compound library from the ZINC database [199]. High-confidence hits with favorable predicted PE values were prioritized, and compounds further filtered by unsatisfied hydrogen bonds. As a result, three promising candidate molecules were identified for potential PIN1 inhibition. These candidates merit experimental validation to assess biochemical efficacy and selectivity.

6.2 Materials and Methods

6.2.1 Datasets

We curated and processed data from the PubChem BioAssay database, focusing on Assay ID 504891. This quantitative high-throughput screening (qHTS) assay used a fluorescence polarization–based competitive binding format to identify small-molecule inhibitors of PIN1. Compounds were evaluated for their ability to displace a fluorescent peptide probe from the active site of recombinant human PIN1, with reduced FP signal indicating potential inhibition. The dataset includes two primary types of labels derived from the assay: potency and efficacy. Potency is defined as the concentration at which a compound exhibits half-maximal activity, while efficacy represents the maximal observed activity as a percentage of the control. Based on the processed qHTS dataset, we defined a predictive target tailored to the evaluation of small-molecule inhibitors against PIN1. To capture both the potency and the maximal biological effect of a compound, we introduced a composite metric:

$$PE = \sqrt{\frac{\text{Potency}}{\text{Efficacy}}} \quad (6.1)$$

This formulation integrates the dose required to achieve half-maximal inhibition with the observed maximal effect size, thus prioritizing compounds that are not only pharmacologically potent but also biologically effective. After filtering for molecules with complete potency and efficacy measurements, a curated set of 29,494 compounds was obtained for model development. This PE-based target reflects both pharmacodynamic strength and functional relevance, offering a more informative endpoint for learning and ranking candidate PIN1 inhibitors.

The ZINC database [199] is a large-scale collection of commercially available small molecules for virtual screening. In this study, we utilized a curated subset of 249,455 compounds that satisfy standard drug-likeness and screening criteria. This subset was used as the virtual com-

pound library for scoring and hit identification using the trained PIN1 inhibition model.

6.2.2 PIN1 inhibitor screening

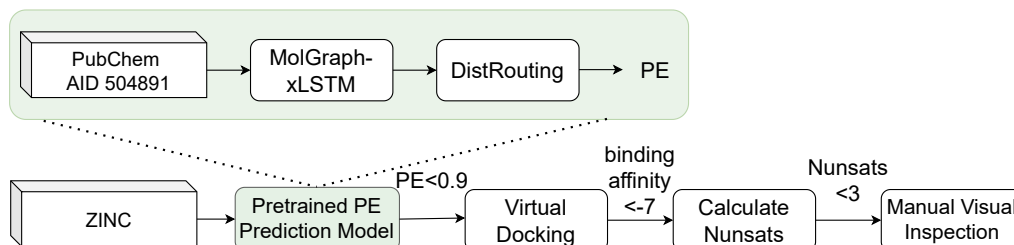


Figure 6.2: **Workflow of PIN1 inhibitor screening.** A PE prediction model trained on PubChem AID 504891 is applied to ZINC compounds, followed by docking, filtering by binding affinity and unsatisfied hydrogen bonds (Nunsats), and final manual inspection.

To identify potential PIN1 inhibitors, we designed a multi-stage virtual screening pipeline combining data-driven prediction and structure-based filtering. As shown in Figure 6.2, a PE prediction model was trained on PubChem AID 504891 using MolGraph-xLSTM for molecular encoding and DistRouting to handle label imbalance across activity ranges.

The trained model was applied to compounds from the ZINC database, and those with predicted $PE < 0.9$ were selected for virtual docking against the PIN1 binding site. Candidates with docking scores below -7 kcal/mol were retained. To ensure binding plausibility, compounds with more than three unsatisfied hydrogen bonds (Nunsats) were excluded. The remaining hits underwent manual inspection to confirm favorable poses and chemical feasibility, resulting in a refined list of high-confidence PIN1 inhibitor candidates.

6.3 Experiment

6.3.1 Experiment Setting

For training the PE prediction model, the dataset derived from PubChem AID 504891 was randomly split into training, validation, and test subsets in a ratio of 8:1:1. Model training

was monitored using the validation set, and the best-performing checkpoint, as measured by validation loss, was selected for final evaluation. The selected model was then assessed on the held-out test set to report generalization performance.

Virtual docking was conducted using AutoDock Vina [18] with the human PIN1 crystal structure (PDB ID: 3IKG) as the receptor. All compounds passing the activity prediction threshold were docked into the binding pocket, and their binding affinities were computed. Subsequently, the number of unsatisfied hydrogen bonds (*Nunsats*) for each protein–ligand complex was calculated using the PLIP (Protein–Ligand Interaction Profiler) [200] to assess interaction quality. These evaluations were used to filter out structurally unstable binding modes and retain only high-confidence candidates.

6.3.2 Screening Results

After training, the PE prediction model was evaluated on the held-out test set. On the PIN1 dataset, it achieved a root mean squared error (RMSE) of 0.441, a Pearson correlation of 0.43, and an AUROC of 0.70 under a 0.9 threshold. These results reflect reasonable predictive power and support its use in screening large chemical libraries.

Table 6.1: Final candidate compounds for PIN1 inhibition after virtual screening.

ID	SMILES	Docking Score (kcal/mol)
mol1	<chem>Cc1ccc(C)c(-c2nn(-c3ccccc3)cc2C=C2\SC(=S)N(C)C2=O)c1</chem>	-7.4
mol2	<chem>O=C1/C(=C/c2csc([N+](=O)[O-])c2)SC(=S)N1c1ccc(F)cc1F</chem>	-7.1
mol3	<chem>O=C1NC([S-])=N/C1=C\c1cccn1-c1ccc([N+](=O)[O-])cc1</chem>	-7.0

When applied to the ZINC database, the model identified 3,536 compounds with predicted PE scores under 0.9. After docking and interaction quality filtering, 53 compounds remained that satisfied both the binding affinity threshold ($< -7\text{kcal/mol}$) and the interaction quality requirement ($Nunsats < 3$). Manual inspection of the docking poses, focusing on alignment within binding pockets and key molecular interactions, further narrowed the list to three final candidate compounds with favorable binding characteristics for PIN1 (Table 6.1).

6.4 Discussion

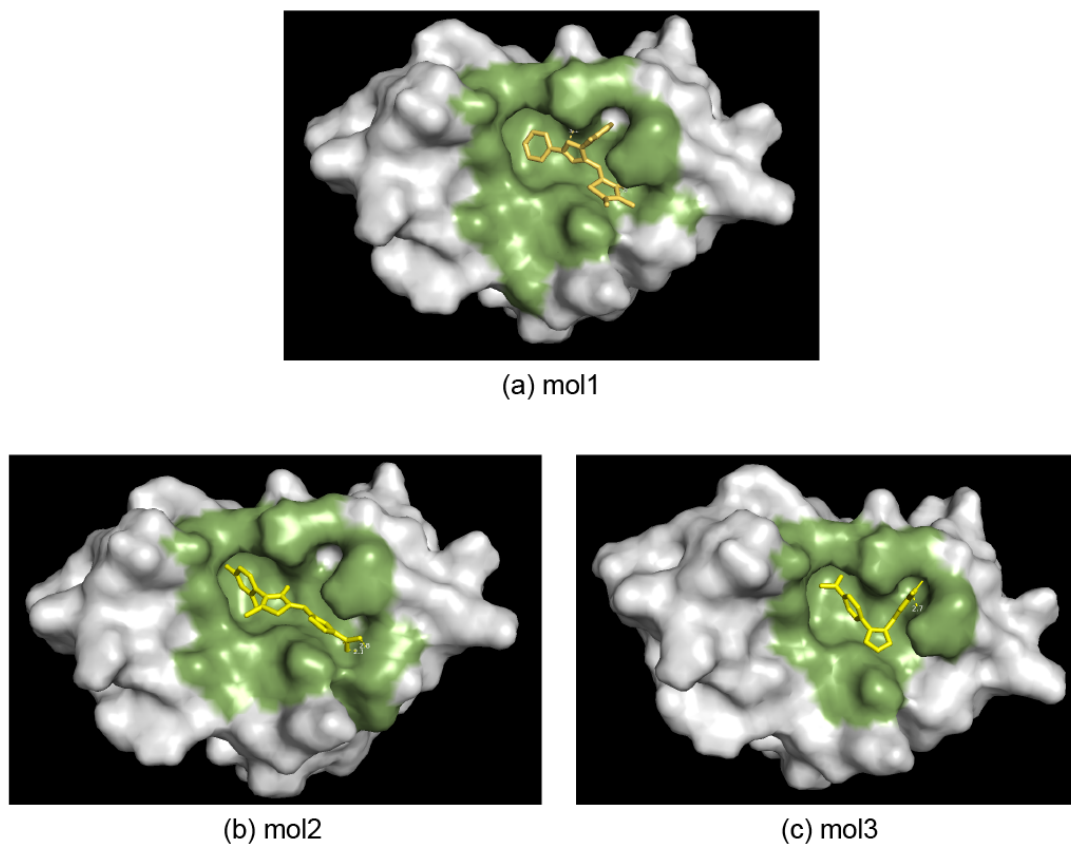


Figure 6.3: Predicted binding poses of top candidate molecules within the PIN1 active site.

The inhibitor binding site of PIN1 comprises three structurally and functionally distinct subregions. The first includes Lys63, Arg68, and Arg69, forming the canonical phosphate binding loop (PBL), which anchors phosphorylated substrates through electrostatic and hydrogen bonding interactions. The second is an aromatic-rich hydrophobic cavity composed of residues such as His59, Met130, Phe134, and His157, facilitating π - π stacking and dispersion interactions. The third is a more flexible and solvent-exposed area involving His59, Ala118, and Leu122, which may accommodate ligand extension and contribute to overall binding stability [187, 194, 201].

Binding analysis of the three candidate compounds is shown in Figure 6.3 and Figure 6.4. **Mol1** exhibits the most comprehensive interaction profile, simultaneously engaging the phosphate-binding loop and the adjacent hydrophobic cavity. It forms four hydrogen bonds with key

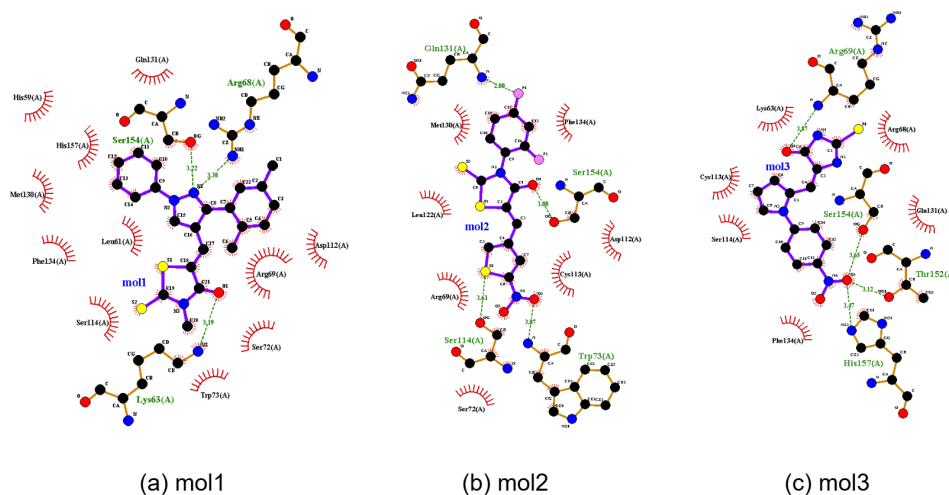


Figure 6.4: 2D interaction diagrams of top candidate molecules with PIN1. The ligand–residue interactions are shown for three candidates. Hydrogen bonds are indicated with green dashed lines, and surrounding residues involved in hydrophobic contacts are labeled with red arcs.

residues including Lys63, Arg68, and Arg69, and establishes multiple π –cation interactions with Arg68, Arg69, and His157. Additionally, π –stacking interactions with Phe134 support aromatic anchoring within the hydrophobic region. These multipoint contacts suggest a stable and high-affinity binding conformation.

Mol2 shows a more limited interaction pattern. It forms a π –cation interaction with Arg69, a halogen bond with Gln131, and a π –stacking interaction with Phe134. However, it does not form hydrogen bonds with Lys63 or Arg68 and lacks contact with the phosphate-recognition residues, suggesting weak anchoring to the polar core and minimal engagement with peripheral regions.

Mol3 mainly interacts with the polar residues surrounding the phosphate-binding loop. It forms hydrogen bonds with Lys63, Arg69, and Ser154, and π –cation interactions with Arg68 and Arg69. Despite lacking classical acidic groups, it maintains strong electrostatic and polar interactions with positively charged residues. However, no contact was observed with hydrophobic or distal residues, indicating a spatially restricted binding mode.

Overall, **mol1** demonstrates the strongest and most diverse interaction profile, combining polar and hydrophobic contacts across the active site. **Mol3** exhibits focused polar interactions

but lacks hydrophobic engagement, while **mol2** shows moderate hydrophobic interactions with limited polar anchoring. These findings highlight the advantage of spatially distributed interactions in promoting binding stability and specificity.

6.5 Summary

In this case study, we presented a hybrid AI-driven virtual screening pipeline for discovering small-molecule inhibitors of PIN1, a master regulator implicated in triple-negative breast cancer (TNBC). By integrating MolGraph-xLSTM (Chapter 3) for molecular encoding and DistRouting (Chapter 4) for addressing target imbalance, we trained a predictive model on a curated fluorescence polarization assay dataset (PubChem AID 504891) using a biologically informed composite metric, PE, that combines potency and efficacy.

The trained model was applied to a subset of 249,455 compounds from the ZINC database to identify candidates with high predicted activity. Compounds that passed the thresholds for predicted efficacy and docking score were further filtered based on interaction quality, using the number of unsatisfied hydrogen bonds and visual inspection. This workflow resulted in the identification of three promising inhibitors with favorable docking scores and binding profiles.

Interaction analysis revealed that the top candidate (mol1) simultaneously engaged polar and hydrophobic subregions of the PIN1 catalytic site, suggesting strong binding affinity. The results demonstrate the potential of combining geometric deep learning and interpretable filtering criteria for efficient prioritization of candidate compounds. This preliminary study lays the groundwork for more comprehensive, mechanism-aware virtual screening systems that incorporate biological context into early-stage drug discovery.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

This thesis presents a comprehensive study on geometric deep learning methods for molecular representation, with a particular emphasis on their applications in drug discovery. The work revolves around four core contributions, namely BSCL (Chapter 2), MolGraph-xLSTM (Chapter 3), DistRouting (Chapter 4), and iNGNN-DTI (Chapter 5).

BSCL proposes a bimodal contrastive learning framework that integrates molecular graphs and SMILES, using a weighted loss to handle regression tasks with limited exact label matches. MolGraph-xLSTM combines atom-level GCN with xLSTM for long-range dependencies, and applies xLSTM to motif-level graphs, with MHMoE enabling adaptive expert routing for improved accuracy and interpretability. DistRouting addresses the challenge of label imbalance commonly found in molecular datasets, proposing a distribution-aware Mixture-of-Experts mechanism that adaptively routes information based on auxiliary descriptors. iNGNN-DTI offers an interpretable geometric model for drug–target interaction prediction.

Overall, the proposed methods introduce several geometric deep learning–based frameworks, highlighting geometric deep learning as a key approach for advancing molecular representation and driving applications in drug discovery.

7.2 Future Work

Looking ahead, there are several promising directions for extending the work presented in this thesis, particularly toward more biologically grounded and mechanistically informed virtual screening systems. While the proposed models primarily focus on molecular structure and property prediction, real-world drug efficacy also depends on how compounds interact with complex biological systems.

One direction is the development of intelligent evaluation agents that integrate multi-omics data, biological knowledge, and pharmacokinetic properties to provide more holistic assessments of candidate compounds. Current virtual screening methods often overlook downstream effects, such as changes in gene expression, metabolic pathways, or cellular phenotypes. By incorporating transcriptomic, proteomic, and metabolomic data, it becomes possible to estimate the biological consequences of molecular interventions. In addition, knowledge mining can link compounds to known mechanisms of action, off-targets, and supporting literature, offering a more interpretable basis for compound selection.

Another direction involves building mechanistic reasoning agents that simulate causal relationships between drugs, targets, signaling pathways, and system-level outcomes. These agents, akin to virtual cells, could trace compound–target interactions through curated biological pathways from databases such as KEGG or Reactome to assess whether a drug acts through therapeutically relevant mechanisms. This mechanistic understanding is essential, especially for diseases requiring precise pathway modulation or prone to severe side effects.

By combining multi-omics-guided evaluation with mechanistic simulation, future AI systems can move beyond structure-based screening toward a more integrated and biologically contextualized framework. These models would not only predict which compounds are likely to be effective but also explain how and why they work, enhancing transparency and confidence in experimental validation. Ultimately, such advances would move the field closer to intelligent, interpretable, and clinically meaningful AI-driven drug discovery.

References

- [1] Siyu He, Linjie Li, Rui Jin, and Xiaojie Lu. Biological function of pin1 in vivo and its inhibitors for preclinical study: early development, current strategies, and future directions. *Journal of Medicinal Chemistry*, 66(14):9251–9277, 2023.
- [2] Sandeep Sinha and Divya Vohora. Drug discovery and development: An overview. In *Pharmaceutical Medicine and Translational Clinical Research*, pages 19–58. Elsevier, 2018.
- [3] Jessica Vamathevan, Dominic Clark, Paul Czodrowski, Ian Dunham, Edgardo Ferran, George Lee, Bin Li, Anant Madabhushi, Parantu Shah, Michaela Spitzer, et al. Applications of machine learning in drug discovery and development. *Nature reviews Drug discovery*, 18(6):463–477, 2019.
- [4] Christopher M. Dobson. Chemical space and biology. *Nature*, 432(7019):824–828, December 2004.
- [5] Chrysanthos Stergiopoulos, Fotios Tsopelas, and Klara Valko. Prediction of herg inhibition of drug discovery compounds using biomimetic hplc measurements. *ADMET & DMPK*, 9(3):191–207, 2021.
- [6] MJ Waring and et al. An analysis of the attrition of drug candidates from four major pharmaceutical companies. *Nature Reviews Drug Discovery*, 14:475–486, 2015.

- [7] David C. Swinney and Jason Anthony. Phenotypic vs. target-based drug discovery for first-in-class medicines. *Nature Reviews Drug Discovery*, 10(7):507–519, 2011.
- [8] Yehudit Hasin, Marcus Seldin, and Aldons Lusi. Multi-omics approaches to disease. *Genome Biology*, 18(1):83, 2017.
- [9] Uros Kuzmanov and Andrew Emili. Protein–protein interaction networks: probing disease mechanisms using model systems. *Genome medicine*, 5(4):37, 2013.
- [10] Milagros Marín, Francisco J Esteban, Hilario Ramírez-Rodrigo, Eduardo Ros, and María José Sáez-Lara. An integrative methodology based on protein–protein interaction networks for identification and functional annotation of disease-relevant genes applied to channelopathies. *BMC bioinformatics*, 20(1):565, 2019.
- [11] Georgios A Pavlopoulos, Maria Secier, Constantin N Moschopoulos, Theodoros G Soldatos, Sophia Kossida, Jan Aerts, Reinhard Schneider, and Pantelis G Bagos. Using graph theory to analyze biological networks. *BioData mining*, 4(1):10, 2011.
- [12] Arwa Raies, Ewa Tulodziecka, James Stainer, Lawrence Middleton, Ryan S Dhindsa, Pamela Hill, Ola Engkvist, Andrew R Harper, Slavé Petrovski, and Dimitrios Vitsios. Drugnomeai is an ensemble machine-learning framework for predicting druggability of candidate drug targets. *Communications Biology*, 5(1):1291, 2022.
- [13] Phasit Charoenkwan, Nalini Schaduangrat, Pietro Lio’, Mohammad Ali Moni, Watshara Shoombuatong, and Balachandran Manavalan. Computational prediction and interpretation of druggable proteins using a stacked ensemble-learning framework. *iScience*, 25(9):104883, 2022.
- [14] Jason D Ulrich and David M Holtzman. Trem2 function in alzheimer’s disease and neurodegeneration. *ACS Chemical Neuroscience*, 7(4):420–427, 2017.

- [15] Brian K Shoichet. Virtual screening of chemical libraries. *Nature*, 432(7019):862–865, 2004.
- [16] Hongbin Li, Yunyun Gao, Yan Cui, and Zhi Li. Machine learning and deep learning in virtual screening: advances, challenges and perspectives. *Drug Discovery Today*, 27(4):1030–1044, 2022.
- [17] Guangfeng Zhou, Domnita-Valeria Rusnac, Hahnbeom Park, et al. An artificial intelligence accelerated virtual screening platform for drug discovery. *Nature Communications*, 15(1):7761, 2024.
- [18] Oleg Trott and Arthur 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.
- [19] Richard A Friesner, Jay L Banks, Robert B Murphy, Thomas A Halgren, Jasna J Klicic, Daniel T Mainz, Matthew P Repasky, Eric H Knoll, Matthew Shelley, Jeffrey K Perry, et al. Glide: a new approach for rapid, accurate docking and scoring. 1. method and assessment of docking accuracy. *Journal of Medicinal Chemistry*, 47(7):1739–1749, 2004.
- [20] Gareth Jones, Peter Willett, Robert C Glen, Andrew R Leach, and Robin Taylor. Development and validation of a genetic algorithm for flexible docking. *Journal of Molecular Biology*, 267(3):727–748, 1997.
- [21] Kumarappan Chidambaram. Identification of bace-1 inhibitors against alzheimer’s disease through e-pharmacophore-based virtual screening and molecular dynamics simulation studies: An in silico approach. *Life*, 13(4):952, 2023.
- [22] Hye Jin Shin, Mi-Hwa Kim, Joo-Youn Lee, Insu Hwang, Gun Young Yoon, Hae Soo Kim, Young-Chan Kwon, Dae-Gyun Ahn, Kyun-Do Kim, Bum-Tae Kim, Seong-Jun

- Kim, and Chonsaeng Kim. Structure-based virtual screening: Identification of a novel ns2b-ns3 protease inhibitor with potent antiviral activity against zika and dengue viruses. *Microorganisms*, 9(3):545, 2021.
- [23] Qinfang Sun, Avik Biswas, RSK Vijayan, Pierrick Craveur, Stefano Forli, Arthur J Olson, Andres Emanuelli Castaner, Karen A Kirby, Stefan G Sarafianos, Nanjie Deng, et al. Structure-based virtual screening workflow to identify antivirals targeting hiv-1 capsid. *Journal of Computer-Aided Molecular Design*, 36(3):193–203, 2022.
- [24] Shan Huang and Xiaoqin Zou. Advances in protein–ligand docking: benchmarking and drug discovery applications. *Current Opinion in Structural Biology*, 20(2):178–186, 2010.
- [25] Eugen Lounkine, Michael J Keiser, Steven Whitebread, Dmitry Mikhailov, Jacques Hamon, Jeremy L Jenkins, Paul Lavan, Eugene Weber, Alexandre Doak, Serge Cote, et al. Large-scale prediction and testing of drug activity on side-effect targets. *Nature*, 486(7403):361–367, 2012.
- [26] Jin Li, Ailing Fu, and Le Zhang. An overview of scoring functions used for protein–ligand interactions in molecular docking. *Interdisciplinary Sciences: Computational Life Sciences*, 11:320–328, 2019.
- [27] Sascha Jung, Helge Vatheuer, and Paul Czodrowski. Vsflow: an open-source ligand-based virtual screening tool. *Journal of Cheminformatics*, 15(1):40, 2023.
- [28] Ali Ezzat, Min Wu, Xiao-Li Li, and Chee-Keong Kwoh. Computational prediction of drug–target interactions using chemogenomic approaches: an empirical survey. *Briefings in Bioinformatics*, 20(4):1337–1357, 2019.
- [29] Tong He, Marten Heidemeyer, Fuqiang Ban, Artem Cherkasov, and Martin Ester. Simboost: a read-across approach for predicting drug–target binding affinities using gradient boosting machines. *Journal of cheminformatics*, 9:1–14, 2017.

- [30] Yuanfang Liu, Min Wu, Chunyu Miao, Peng Zhao, Xiaoyan Li, Zhuo Liu, Bo Xu, Guangfu Pei, Xiang Zhang, Yuanyuan Li, et al. Dtinet: a network integration approach for drug–target interaction prediction and computational drug repositioning from heterogeneous information. *Nature Communications*, 7:10325, 2016.
- [31] Christopher A Lipinski. Lead- and drug-like compounds: the rule-of-five revolution. *Drug Discovery Today: Technologies*, 1(4):337–341, 2004.
- [32] David Lagorce, Olivier Sperandio, Jonathan B Baell, Maria A Miteva, and Bruno O Villoutreix. Faf-drugs2: free adme/tox filtering tool to assist drug discovery and chemical biology projects. *BMC bioinformatics*, 9(1):396, 2008.
- [33] Gabriele Cruciani, Manuel Pastor, and Wolfgang Guba. Volsurf: a new tool for the pharmacokinetic optimization of lead compounds. *European Journal of Pharmaceutical Sciences*, 11:S29–S39, 2000.
- [34] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.
- [35] Andreas Mayr, Günter Klambauer, Thomas Unterthiner, and Sepp Hochreiter. Deeptox: toxicity prediction using deep learning. *Frontiers in Environmental Science*, 3:80, 2016.
- [36] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.
- [37] Hakime Ozturk, Elif Ozkirimli, and Arzucan Ozgur. Deepdta: deep drug–target binding affinity prediction. *Bioinformatics*, 34(17):i821–i829, 2018.
- [38] In Lee, Juhyun Keum, and Hojung Nam. Deepconv-dti: Prediction of drug–target interactions via deep learning with convolution on protein sequences. *PLoS computational biology*, 15(6):e1007129, 2019.

- [39] Ziduo Yang, Weihe Zhong, Lu Zhao, and Calvin Yu-Chian Chen. Ml-dti: mutual learning mechanism for interpretable drug–target interaction prediction. *The Journal of Physical Chemistry Letters*, 12(17):4247–4261, 2021.
- [40] Kexin Huang, Cao Xiao, Lucas M Glass, and Jimeng Sun. Moltrans: Molecular interaction transformer for drug–target interaction prediction. *Bioinformatics*, 37(6):830–836, 2021.
- [41] Mostafa Karimi, Di Wu, Zhe Wang, and Yizhou Shen. Deepaffinity: interpretable deep learning of compound–protein affinity through unified recurrent and convolutional neural networks. *Bioinformatics*, 35(18):3329–3338, 2019.
- [42] Jing Tang, Andrzej Szwejda, Sushant Shakyawar, Tao Xu, Pauliina Hintsanen, Krister Wennerberg, and Tero Aittokallio. Making sense of large-scale kinase inhibitor bioactivity data sets: a comparative and integrative analysis. *Journal of Chemical Information and Modeling*, 54(3):735–743, 2014.
- [43] Mindy I Davis, Jeremy P Hunt, Sanna Herrgard, Pietro Cicceri, Lisa M Wodicka, Gabriel Pallares, Michael Hocker, Daniel K Treiber, and Patrick P Zarrinkar. Comprehensive analysis of kinase inhibitor selectivity. *Nature biotechnology*, 29(11):1046–1051, 2011.
- [44] Marwin HS Segler, Thierry Kogej, Christian Tyrchan, and Mark P Waller. Generating focused molecule libraries for drug discovery with recurrent neural networks. *Nature*, 555(7698):604–610, 2018.
- [45] Alex Zhavoronkov, Yan A Ivanenkov, Alex Aliper, Mark S Veselov, Vladimir A Aladinskiy, Alena V Aladinskaya, Vasily A Terentiev, Daniil A Polykovskiy, Mikhail D Kuznetsov, Arman Asadulaev, et al. Deep learning enables rapid identification of potent ddr1 kinase inhibitors. *Nature Biotechnology*, 37(9):1038–1040, 2019.
- [46] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna

- Potapenko, et al. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021.
- [47] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.
- [48] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 1263–1272. PMLR, 2017.
- [49] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.
- [50] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, et al. Graph attention networks. *stat*, 1050(20):10–48550, 2017.
- [51] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015.
- [52] Guohao Li, Matthias Müller, Ali Thabet, and Bernard Ghanem. Deepgcns: Can gcns go as deep as cnns? In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9267–9276, 2019.
- [53] Gabriele Corso, Fabrizio Cavalleri, Dominique Beaini, Pietro Liò, and Petar Veličković. Principal neighbourhood aggregation for graph nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 13260–13271, 2020.
- [54] Garrett B Goh, Charles Siegel, and Abhinav Vishnu. Smiles2vec: An interpretable general-purpose deep neural network for predicting chemical properties. *arXiv preprint arXiv:1712.02034*, 2017.

- [55] Andreas Mayr, Günter Klambauer, Thomas Unterthiner, and Sepp Hochreiter. Large-scale comparison of machine learning methods for drug toxicity prediction. *Journal of cheminformatics*, 7(1):1–14, 2015.
- [56] Seonwoo Min, Byunghan Lee, and Sungroh Yoon. Deep learning in bioinformatics. *Briefings in bioinformatics*, 18(5):851–869, 2017.
- [57] Yongqing Zhang, Shaojie Qiao, Shengjie Ji, and Yizhou Li. Deepsite: bidirectional lstm and cnn models for predicting dna–protein binding. *International Journal of Machine Learning and Cybernetics*, 11:841–851, 2020.
- [58] Yifan Wang and et al. Deepttc: a deep neural network model for predicting drug–target interactions based on integrated information. In *Briefings in bioinformatics*, volume 23, 2022.
- [59] Mengdi Wen, Mingjun Zhang, Peilin Zhao, and Jinfeng Li. Fingerprints are all you need: A fingerprint-based deep learning method for molecular property prediction. *Journal of Chemical Information and Modeling*, 62(13):3084–3093, 2022.
- [60] Pavel Karpov, Guillaume Godin, and Igor V Tetko. A transformer-cnn neural network model for qsar modeling. *Journal of Cheminformatics*, 12(1):1–11, 2020.
- [61] Kevin Yang, Linus Swanson, Wengong Jin, and et al. Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, 59(8):3370–3388, 2019.
- [62] Zhaoping Xiong, Dingyan Wang, Xiaohong Liu, Feisheng Zhong, Xiaozhe Wan, Xutong Li, Zhaojun Li, Xiaomin Luo, Kaixian Chen, Hualiang 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.

- [63] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems (NeurIPS)*, 30, 2017.
- [64] Thin Nguyen, Hang Le, Thomas P Quinn, Tri Nguyen, Thuc Duy Le, and Svetha Venkatesh. Graphdta: predicting drug–target binding affinity with graph neural networks. *Bioinformatics*, 37(8):1140–1147, 2021.
- [65] Ziduo Yang, Weihe Zhong, Lu Zhao, and Calvin Yu-Chian Chen. Mgraphdta: deep multiscale graph neural network for explainable drug–target binding affinity prediction. *Chemical science*, 13(3):816–833, 2022.
- [66] Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Saucedo Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in neural information processing systems (NeurIPS)*, 30, 2017.
- [67] Johannes Gasteiger, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. In *International Conference on Learning Representations*, pages 1–13, 2020.
- [68] Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR, 2021.
- [69] Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. Se (3)-transformers: 3d roto-translation equivariant attention networks. *Advances in neural information processing systems (NeurIPS)*, 33:1970–1981, 2020.
- [70] Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael JL Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons. *arXiv preprint arXiv:2009.01411*, 2020.

- [71] Fangfang Ren, Tianyu Zhao, Jie Zhang, Yufei Wang, et al. A small-molecule tnik inhibitor targets fibrosis in preclinical and clinical models. *Nature Biotechnology*, 43:63–75, 2024.
- [72] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.
- [73] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. In *Journal of Machine Learning Research*, volume 21, pages 1–67, 2020.
- [74] Alexander Rives, Joshua Meier, Tom Sercu, and et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15), 2021.
- [75] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, and et al. Prottrans: Towards cracking the language of life’s code through self-supervised deep learning and high performance computing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
- [76] Benjamin Fabian, Jonas Edlich, Joerg Rodde, and et al. Molecular representation learning with language models and domain-relevant auxiliary tasks. *arXiv preprint arXiv:2011.13230*, 2020.
- [77] Sambhavi Chithrananda, Gabriel Grand, and Bharath Ramsundar. Chemberta: Large-scale self-supervised pretraining for molecular property prediction. *arXiv preprint arXiv:2010.09885*, 2020.

- [78] Sheng Wang, Yuzhe Guo, Zhengyang Wang, and et al. Smiles-bert: Large scale unsupervised pre-training for molecular property prediction. *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, pages 429–436, 2019.
- [79] Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. Self-supervised graph transformer on large-scale molecular data. *Advances in neural information processing systems (NeurIPS)*, 33:12559–12571, 2020.
- [80] Yuyang Wang, Jianren Wang, Zhonglin Cao, and Amir Barati Farimani. Molecular contrastive learning of representations via graph neural networks. *Nature Machine Intelligence*, 4(3):279–287, 2022.
- [81] Kun Zhou, Yutong Gong, Tong Xie, and et al. Uni-mol: A universal 3d molecular representation learning framework. *Nature Communications*, 14(1):2573, 2023.
- [82] Zhaohong Wu, Zhichao Li, Chang-Yu Hsieh, and et al. Molformer: Molecular representation learning via transformer-based architecture. *arXiv preprint arXiv:2202.05150*, 2022.
- [83] Xiaomin Fang, Lihang Liu, Jieqiong Lei, Donglong He, Shanzhuo Zhang, Jingbo Zhou, Fan Wang, Hua Wu, and Haifeng Wang. Geometry-enhanced molecular representation learning for property prediction. *Nature Machine Intelligence*, 4(2):127–134, 2022.
- [84] Hannes Stärk, Octavian Ganea, Lagnajit Pattanaik, Regina Barzilay, and Tommi Jaakkola. Equibind: Geometric deep learning for drug binding structure prediction. In *International Conference on Machine Learning*, pages 20503–20521. PMLR, 2022.
- [85] Yan Sun, Mohaiminul Islam, Ehsan Zahedi, Mélaïne Kuenemann, Hassan Chouaib, and Pingzhao Hu. Molecular property prediction based on bimodal supervised contrastive learning. In *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 394–397. IEEE, 2022.

- [86] Yan Sun, Yan Yi Li, Carson K Leung, and Pingzhao Hu. ingnn-dti: prediction of drug–target interaction with interpretable nested graph neural network and pretrained molecule models. *Bioinformatics*, 40(3):btae135, 2024.
- [87] Oliver Wieder, Stefan Kohlbacher, Méline Kuenemann, Arthur Garon, Pierre Ducrot, Thomas Seidel, and Thierry Langer. A compact review of molecular property prediction with graph neural networks. *Drug Discovery Today: Technologies*, 37:1–12, 2020.
- [88] W Patrick Walters and Regina Barzilay. Applications of deep learning in molecule generation and molecular property prediction. *Accounts of chemical research*, 54(2):263–270, 2020.
- [89] Rémy Brossard, Oriel Frigo, and David Dehaene. Graph convolutions that can finally model local structure. *arXiv preprint arXiv:2011.15069*, 2020.
- [90] Guohao Li, Chenxin Xiong, Guocheng Qian, Ali Thabet, and Bernard Ghanem. Deep-ergcn: training deeper gcns with generalized aggregation functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):13024–13034, 2023.
- [91] Vishaal Udandara, Abhishek Maiti, Deepak Srivatsav, Suryatej Reddy Vyalla, Yifang Yin, and Rajiv Ratn Shah. Cobra: Contrastive bi-modal representation algorithm. *arXiv preprint arXiv:2005.03687*, 2020.
- [92] Mohammadreza Zolfaghari, Yi Zhu, Peter Gehler, and Thomas Brox. Crossclr: Cross-modal contrastive learning for multi-modal video representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1450–1459, 2021.
- [93] Han Zhang, Jing Yu Koh, Jason Baldridge, Honglak Lee, and Yinfei Yang. Cross-modal contrastive learning for text-to-image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 833–842, 2021.

- [94] Xin Yuan, Zhe Lin, Jason Kuen, Jianming Zhang, Yilin Wang, Michael Maire, Ajinkya Kale, and Baldo Faieta. Multimodal contrastive training for visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6995–7004, 2021.
- [95] Devamanyu Hazarika, Roger Zimmermann, and Soujanya Poria. Misa: Modality-invariant and-specific representations for multimodal sentiment analysis. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1122–1131, 2020.
- [96] Mengying Sun, Jing Xing, Huijun Wang, Bin Chen, and Jiayu Zhou. Mocl: Contrastive learning on molecular graphs with multi-level domain knowledge. *arXiv preprint arXiv:2106.04509*, 2021.
- [97] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems (NeurIPS)*, 33:18661–18673, 2020.
- [98] Jinhua Zhu, Yingce Xia, Lijun Wu, Shufang Xie, Wengang Zhou, Tao Qin, Houqiang Li, and Tie-Yan Liu. Dual-view molecular pre-training. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3615–3627, 2023.
- [99] Shuangli Li, Jingbo Zhou, Tong Xu, Dejing Dou, and Hui Xiong. Geomgcl: Geometric graph contrastive learning for molecular property prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 4541–4549, 2022.
- [100] Joseph L Durant, Burton A Leland, Douglas R Henry, and James G Nourse. Reoptimization of mdl keys for use in drug discovery. *Journal of chemical information and computer sciences*, 42(6):1273–1280, 2002.
- [101] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50(5):742–754, 2010.

- [102] Greg Landrum. RDKit: Open-source cheminformatics, 2006. <https://www.rdkit.org>.
- [103] Zhengyang Wang, Meng Liu, Youzhi Luo, Zhao Xu, Yaochen Xie, Limei Wang, Lei Cai, and Shuiwang Ji. Moleculekit: Machine learning methods for molecular property prediction and drug discovery. *arXiv preprint arXiv:2012.01981*, 2020.
- [104] John J Irwin and Brian K Shoichet. Zinc- a free database of commercially available compounds for virtual screening. *Journal of chemical information and modeling*, 45(1):177–182, 2005.
- [105] Xiaomin Fang, Lihang Liu, Jieqiong Lei, Donglong He, Shanzhuo Zhang, Jingbo Zhou, Fan Wang, Hua Wu, and Haifeng Wang. Geometry-enhanced molecular representation learning for property prediction. *Nature Machine Intelligence*, 4(2):127–134, 2022.
- [106] Denise B Catacutan, Jeremie Alexander, Autumn Arnold, and Jonathan M Stokes. Machine learning in preclinical drug discovery. *Nature Chemical Biology*, 20(8):960–973, 2024.
- [107] Lei Jia and Hua Gao. Machine learning for in silico admet prediction. *Artificial Intelligence in Drug Design*, pages 447–460, 2022.
- [108] José Jiménez-Luna, Francesca Grisoni, and Gisbert Schneider. Drug discovery with explainable artificial intelligence. *Nature Machine Intelligence*, 2(10):573–584, 2020.
- [109] Anastasiia V Sadybekov and Vsevolod Katritch. Computational approaches streamlining drug discovery. *Nature*, 616(7958):673–685, 2023.
- [110] Zhaoping Xiong, Dingyan Wang, Xiaohong Liu, Feisheng Zhong, Xiaozhe Wan, Xutong Li, Zhaojun Li, Xiaomin Luo, Kaixian Chen, Hualiang 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.

- [111] Hanxuan Cai, Huimin Zhang, Duancheng Zhao, Jingxing Wu, and Ling Wang. Fp-gnn: a versatile deep learning architecture for enhanced molecular property prediction. *Briefings in bioinformatics*, 23(6):bbac408, 2022.
- [112] Jian Gao, Zheyuan Shen, Yufeng Xie, Jialiang Lu, Yang Lu, Sikang Chen, Qingyu Bian, Yue Guo, Liteng Shen, Jian Wu, et al. Transfoxmol: predicting molecular property with focused attention. *Briefings in Bioinformatics*, 24(5):bbad306, 2023.
- [113] Xuan Zang, Xianbing Zhao, and Buzhou Tang. Hierarchical molecular graph self-supervised learning for property prediction. *Communications Chemistry*, 6(1):34, 2023.
- [114] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Message passing neural networks. In *Machine learning meets quantum physics*, pages 199–214. Springer, 2020.
- [115] Patrick Reiser, Marlen Neubert, André Eberhard, Luca Torresi, Chen Zhou, Chen Shao, Houssam Metni, Clint van Hoesel, Henrik Schopmans, Timo Sommer, et al. Graph neural networks for materials science and chemistry. *Communications Materials*, 3(1):93, 2022.
- [116] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.
- [117] Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. *arXiv preprint arXiv:2006.05205*, 2020.
- [118] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [119] Maximilian Beck, Korbinian Pöppel, Markus Spanring, Andreas Auer, Oleksandra Prudnikova, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp

- Hochreiter. xlstm: Extended long short-term memory. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:107547–107603, 2025.
- [120] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems (NeurIPS)*, 30, 2017.
- [121] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.
- [122] Peter Ertl. An algorithm to identify functional groups in organic molecules. *Journal of cheminformatics*, 9:1–7, 2017.
- [123] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In *International conference on machine learning*, pages 5453–5462. PMLR, 2018.
- [124] Shaohan Huang, Xun Wu, Shuming Ma, and Furu Wei. Mh-moe: Multi-head mixture-of-experts. *arXiv preprint arXiv:2411.16205*, 2024.
- [125] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.
- [126] Weimin Zhu, Yi Zhang, Duancheng Zhao, Jianrong Xu, and Ling Wang. Hignn: A hierarchical informative graph neural network for molecular property prediction equipped with feature-wise attention. *Journal of Chemical Information and Modeling*, 63(1):43–55, 2022.
- [127] Jorg Degen, Christof Wegscheid-Gerlach, Andrea Zaliani, and Matthias Rarey. On

- the art of compiling and using 'drug-like' chemical fragment spaces. *ChemMedChem*, 3(10):1503, 2008.
- [128] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.
- [129] Zixiang Chen, Yihe Deng, Yue Wu, Quanquan Gu, and Yuanzhi Li. Towards understanding the mixture-of-experts layer in deep learning. *Advances in neural information processing systems (NeurIPS)*, 35:23049–23062, 2022.
- [130] Zewei Ji, Runhan Shi, Jiarui Lu, Fang Li, and Yang Yang. Relmole: Molecular representation learning based on two-level graph similarities. *Journal of Chemical Information and Modeling*, 62(22):5361–5372, 2022.
- [131] Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf Roohani, Jure Leskovec, Connor W Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. *arXiv preprint arXiv:2102.09548*, 2021.
- [132] Erin LeDell and Sebastien Poirier. H2o automl: Scalable automatic machine learning. In *Proceedings of the AutoML Workshop at ICML*, volume 2020, page 24, 2020.
- [133] David Daniel, Stephen Bacchi, Robert Casson, and Weng Chan. Sulfonamides in ophthalmology: adverse reactions: Evidence-based use of sulfa drugs in ophthalmology. *International Ophthalmology*, 44(1):214, 2024.
- [134] Ivan Ivanov and Vincent R. Lee. *Hydrazine Toxicology*. StatPearls Publishing, Treasure Island (FL), 2023.
- [135] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
- [136] Andrew T Placzek, Skylar J Ferrara, Meredith D Hartley, Hannah S Sanford-Crane, J Matthew Meinig, and Thomas S Scanlan. Sobetirome prodrug esters with enhanced

- blood–brain barrier permeability. *Bioorganic & medicinal chemistry*, 24(22):5842–5854, 2016.
- [137] Skylar J Ferrara and Thomas S Scanlan. A cns-targeting prodrug strategy for nuclear receptor modulators. *Journal of Medicinal Chemistry*, 63(17):9742–9751, 2020.
- [138] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.
- [139] Zhi-Hua Zhou and Xu-Ying Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on knowledge and data engineering*, 18(1):63–77, 2006.
- [140] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277, 2019.
- [141] Paula Branco, Luís Torgo, and Rita P Ribeiro. Smogn: a pre-processing approach for imbalanced regression. In *First international workshop on learning with imbalanced domains: Theory and applications*, pages 36–50. PMLR, 2017.
- [142] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in neural information processing systems (NeurIPS)*, 33:5812–5823, 2020.
- [143] Yuzhe Yang, Kaiwen Zha, Yingcong Chen, Hao Wang, and Dina Katabi. Delving into deep imbalanced regression. In *International conference on machine learning*, pages 11842–11851. PMLR, 2021.
- [144] Ruizhi Pu, Gezheng Xu, Ruiyi Fang, Bing-Kun Bao, Charles Ling, and Boyu Wang. Leveraging group classification with descending soft labeling for deep imbalanced re-

- gression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 19978–19985, 2025.
- [145] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarczyk, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.
- [146] Zhengyang Mao, Wei Ju, Siyu Yi, Yifan Wang, Zhiping Xiao, Qingqing Long, Nan Yin, Xinwang Liu, and Ming Zhang. Learning knowledge-diverse experts for long-tailed graph classification. *ACM Transactions on Knowledge Discovery from Data*, 19(2):1–24, 2025.
- [147] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.
- [148] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265*, 2019.
- [149] Ross Irwin, Spyridon Dimitriadis, Jiazhen He, and Esben Jannik Bjerrum. Chemformer: a pre-trained transformer for computational chemistry. *Machine Learning: Science and Technology*, 3(1):015022, 2022.
- [150] Damai Dai, Chengqi Deng, Chenggang Zhao, RX Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Yu Wu, et al. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*, 2024.
- [151] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning (ICML)*, pages 1597–1607. PMLR, 2020.

- [152] Kanica Sachdev and Manoj Kumar Gupta. A comprehensive review of feature based methods for drug target interaction prediction. *Journal of biomedical informatics*, 93:103159, 2019.
- [153] Si-sheng Ou-Yang, Jun-yan Lu, Xiang-qian Kong, Zhong-jie Liang, Cheng Luo, and Hualiang Jiang. Computational drug discovery. *Acta Pharmacologica Sinica*, 33(9):1131–1140, 2012.
- [154] Ming Wen, Zhimin Zhang, Shaoyu Niu, Haozhi Sha, Ruihan Yang, Yonghuan Yun, and Hongmei Lu. Deep-learning-based drug–target interaction prediction. *Journal of proteome research*, 16(4):1401–1409, 2017.
- [155] Ziduo Yang, Weihe Zhong, Lu Zhao, and Calvin Yu-Chian Chen. Ml-dti: mutual learning mechanism for interpretable drug–target interaction prediction. *The Journal of Physical Chemistry Letters*, 12(17):4247–4261, 2021.
- [156] Kexin Huang, Cao Xiao, Lucas M Glass, and Jimeng Sun. Moltrans: molecular interaction transformer for drug–target interaction prediction. *Bioinformatics*, 37(6):830–836, 2021.
- [157] Mingjian Jiang, Zhen Li, Shugang Zhang, Shuang Wang, Xiaofeng Wang, Qing Yuan, and Zhiqiang Wei. Drug–target affinity prediction using graph neural network and contact maps. *RSC advances*, 10(35):20701–20712, 2020.
- [158] Monica Schenone, Vlado Dančák, Bridget K Wagner, and Paul A Clemons. Target identification and mechanism of action in chemical biology and drug discovery. *Nature chemical biology*, 9(4):232–240, 2013.
- [159] Peizhen Bai, Filip Miljković, Bino John, and Haiping Lu. Interpretable bilinear attention network with domain adaptation improves drug–target prediction. *Nature Machine Intelligence*, 5(2):126–136, 2023.

- [160] Muhan Zhang and Pan Li. Nested graph neural networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:15734–15747, 2021.
- [161] Shuangfei Zhai, Walter Talbott, Nitish Srivastava, Chen Huang, Hanlin Goh, Ruixiang Zhang, and Josh Susskind. An attention free transformer. *arXiv preprint arXiv:2105.14103*, 2021.
- [162] Marinka Zitnik, Rok Sosič, Sagar Maheshwari, and Jure Leskovec. BioSNAP Datasets: Stanford biomedical network dataset collection. <http://snap.stanford.edu/biodata>, August 2018.
- [163] Jose M Duarte, Rajagopal Sathyapriya, Henning Stehr, Ioannis Filippis, and Michael Lappe. Optimal contact definition for reconstruction of contact maps. *BMC bioinformatics*, 11:1–10, 2010.
- [164] Jong cheol Jeong, Xiaotong Lin, and Xue-Wen Chen. On position-specific scoring matrix for protein function prediction. *IEEE/ACM transactions on computational biology and bioinformatics*, 8(2):308–315, 2010.
- [165] Robert C Edgar and Serafim Batzoglou. Multiple sequence alignment. *Current opinion in structural biology*, 16(3):368–373, 2006.
- [166] Dexiong Chen, Leslie O’Bray, and Karsten Borgwardt. Structure-aware transformer for graph representation learning. In *International conference on machine learning*, pages 3469–3489. PMLR, 2022.
- [167] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

- [168] Mirco Michel, David Menéndez Hurtado, and Arne Elofsson. Pconsc4: fast, accurate and hassle-free contact predictions. *Bioinformatics*, 35(15):2677–2679, 2019.
- [169] Jaspreet Singh, Thomas Litfin, Jaswinder Singh, Kuldip Paliwal, and Yaoqi Zhou. Spot-contact-lm: improving single-sequence-based prediction of protein contact map using a transformer language model. *Bioinformatics*, 38(7):1888–1894, 2022.
- [170] Tamás Hegedűs, Markus Geisler, Gergely László Lukács, and Bianka Farkas. Ins and outs of alphafold2 transmembrane protein structure predictions. *Cellular and Molecular Life Sciences*, 79(1):73, 2022.
- [171] Oliviero Carugo and Sándor Pongor. A normalized root-mean-square distance for comparing protein three-dimensional structures. *Protein science*, 10(7):1470–1473, 2001.
- [172] Alejandro Castro-Alvarez, Anna M Costa, and Jaume Vilarrosa. The performance of several docking programs at reproducing protein–macrolide-like crystal structures. *Molecules*, 22(1):136, 2017.
- [173] Sambit K Mohanty, Abhishek Satapathy, Machita M Naidu, Sanjay Mukhopadhyay, Shivani Sharma, Lisa M Barton, Edana Stroberg, Eric J Duval, Dinesh Pradhan, Alexander Tzankov, et al. Severe acute respiratory syndrome coronavirus-2 (sars-cov-2) and coronavirus disease 19 (covid-19)—anatomic pathology perspective on current knowledge. *Diagnostic pathology*, 15:1–17, 2020.
- [174] Suresh Kumar. Covid-19: A drug repurposing and biomarker identification by using comprehensive gene-disease associations through protein-protein interaction network analysis. *Preprints*, 2020.
- [175] Ting-Fu Chen, Yu-Chuan Chang, Yi Hsiao, Ko-Han Lee, Yu-Chun Hsiao, Yu-Hsiang Lin, Yi-Chin Ethan Tu, Hsuan-Cheng Huang, Chien-Yu Chen, and Hsueh-Fen Juan. Dockcov2: a drug database against sars-cov-2. *Nucleic Acids Research*, 49(D1):D1152–D1159, 2021.

- [176] Shaopeng Yu, Yao Chen, Yusen Xiang, He Lin, Mengge Wang, Wenbo Ye, Pei Zhang, Hongzhuan Chen, Guoqiang Lin, Yuying Zhu, et al. Pseudoephedrine and its derivatives antagonize wild and mutated severe acute respiratory syndrome-cov-2 viruses through blocking virus invasion and antiinflammatory effect. *Phytotherapy Research*, 35(10):5847–5860, 2021.
- [177] Jie Mei, Yatong Zhou, Xinping Yang, Fan Zhang, Xiufeng Liu, and Boyang Yu. Active components in ephedra sinica stapf disrupt the interaction between ace2 and sars-cov-2 rbd: Potent covid-19 therapeutic agents. *Journal of ethnopharmacology*, 278:114303, 2021.
- [178] Mizuki Yamamoto, Jin Gohda, Ayako Kobayashi, Keiko Tomita, Youko Hirayama, Naohiko Koshikawa, Motoharu Seiki, Kentaro Semba, Tetsu Akiyama, Yasushi Kawaguchi, et al. Metalloproteinase-dependent and tmprss2-independent cell surface entry pathway of sars-cov-2 requires the furin cleavage site and the s2 domain of spike protein. *Mbio*, 13(4):e00519–22, 2022.
- [179] Xiaohuan Wang, Yongkang Chen, Huichun Shi, and Peng Zou. Erythromycin estolate is a potent inhibitor against hcov-oc43 by directly inactivating the virus particle. *Frontiers in Cellular and Infection Microbiology*, 12:905248, 2022.
- [180] Ozlem Yersal and Sabri Barutca. Biological subtypes of breast cancer: Prognostic and therapeutic implications. *World journal of clinical oncology*, 5(3):412, 2014.
- [181] Kathleen M Mahoney, Paul D Rennert, and Gordon J Freeman. Combination cancer immunotherapy and new immunomodulatory targets. *Nature reviews Drug discovery*, 14(8):561–584, 2015.
- [182] Padmanee Sharma and James P Allison. The future of immune checkpoint therapy. *Science*, 348(6230):56–61, 2015.

- [183] Ameet J Sabnis and Trevor G Bivona. Principles of resistance to targeted cancer therapy: Lessons from basic and translational cancer biology. *Trends in Molecular Medicine*, 25(3):185–197, 2019.
- [184] Shawn P Kubli, Thorsten Berger, Daniel V Araujo, Lillian L Siu, and Tak W Mak. Beyond immune checkpoint blockade: emerging immunological strategies. *Nature reviews Drug discovery*, 20(12):899–919, 2021.
- [185] Diwakar R Pattabiraman and Robert A Weinberg. Tackling the cancer stem cells—what challenges do they pose? *Nature reviews Drug discovery*, 13(7):497–512, 2014.
- [186] Brian C Prager, Qian Xie, Shideng Bao, and Jeremy N Rich. Cancer stem cells: the architects of the tumor ecosystem. *Cell Stem Cell*, 24(1):41–53, 2019.
- [187] Xiao Zhen Zhou and Kun Ping Lu. The isomerase pin1 controls numerous cancer-driving pathways and is a unique drug target. *Nature Reviews Cancer*, 16(7):463–478, 2016.
- [188] Kun Ping Lu and Xiao Z Zhou. Pin1-catalyzed conformational regulation after phosphorylation: a distinct checkpoint in cell signaling and drug discovery. *Science Signaling*, 17(784):eadi8743, 2024.
- [189] Kun Ping Lu. Prolyl isomerase pin1 as a molecular target for cancer diagnostics and therapeutics. *Cancer cell*, 4(3):175–180, 2003.
- [190] Shuo Wei, Shingo Kozono, Lev Kats, Morris Nechama, Wenzong Li, Jlenia Guarnerio, Manli Luo, Mi-Hyeon You, Yandan Yao, Asami Kondo, et al. Active pin1 is a key target of all-trans retinoic acid in acute promyelocytic leukemia and breast cancer. *Nature medicine*, 21(5):457–466, 2015.
- [191] Man-Li Luo, Chang Gong, Chun-Hau Chen, Daniel Y Lee, Hai Hu, Pengyu Huang, Yandan Yao, Wenjun Guo, Ferenc Reinhardt, Gerburg Wulf, et al. Prolyl isomerase

- pin1 acts downstream of mir200c to promote cancer stem-like cell traits in breast cancer. *Cancer research*, 74(13):3603–3616, 2014.
- [192] Angela et al. Rustighi. Prolyl-isomerase pin1 controls normal and cancer stem cells of the breast. *EMBO Molecular Medicine*, 6(1):99–119, 2014.
- [193] Min-Liang et al. Luo. The rab2a gtpase is a breast cancer stem-promoting gene that enhances tumorigenesis via activating erk signaling. *Cell Reports*, 11(7):111–124, 2015.
- [194] Benika J Pinch, Zainab M Doctor, Behnam Nabet, Christopher M Browne, Hyuk-Soo Seo, Mikaela L Mohardt, Shingo Kozono, Xiaolan Lian, Theresa D Manz, Yujin Chun, et al. Identification of a potent and selective covalent pin1 inhibitor. *Nature chemical biology*, 16(9):979–987, 2020.
- [195] Elena Campaner, Alessandra Rustighi, Alessandro Zannini, Alberto Cristiani, Silvano Piazza, Yari Ciani, Ori Kalid, Gali Golan, Erkan Baloglu, Sharon Shacham, et al. A covalent pin1 inhibitor selectively targets cancer cells by a dual mechanism of action. *Nature communications*, 8(1):15772, 2017.
- [196] Jonathan D Moore and Alex Potter. Pin1 inhibitors: pitfalls, progress and cellular pharmacology. *Bioorganic & Medicinal Chemistry Letters*, 23(16):4283–4291, 2013.
- [197] Shingo Kozono, Yu-Min Lin, Hyuk-Soo Seo, Benika Pinch, Xiaolan Lian, Chenxi Qiu, Megan K Herbert, Chun-Hau Chen, Li Tan, Ziang Jeff Gao, et al. Arsenic targets pin1 and cooperates with retinoic acid to inhibit cancer-driving pathways and tumor-initiating cells. *Nature communications*, 9(1):3069, 2018.
- [198] National Center for Biotechnology Information. Pubchem bioassay record for aid 504891: qhts assay to find inhibitors of pin1. <https://pubchem.ncbi.nlm.nih.gov/bioassay/504891>, 2025. Source: National Center for Advancing Translational Sciences (NCATS).

- [199] John J Irwin, Teague Sterling, Michael M Mysinger, Erik S Bolstad, and Ryan G Coleman. Zinc: a free tool to discover chemistry for biology. *Journal of Chemical Information and Modeling*, 52(7):1757–1768, 2012.
- [200] Sebastian Salentin, Sven Schreiber, V Joachim Haupt, Melissa F Adasme, and Michael Schroeder. Plip: fully automated protein–ligand interaction profiler. *Nucleic acids research*, 43(W1):W443–W447, 2015.
- [201] Kazuhiro Koikawa, Shin Kibe, Futoshi Suizu, Nobufumi Sekino, Nami Kim, Theresa D Manz, Benika J Pinch, Dipikaa Akshinthala, Ana Verma, Giorgio Gaglia, et al. Targeting pin1 renders pancreatic cancer eradicable by synergizing with immunochemotherapy. *Cell*, 184(18):4753–4771, 2021.