Feature-Based Deep Learning Approaches to Facilitate Drug Discovery

by

Mingyi Xue

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Chemistry)

at the

UNIVERSITY OF WISCONSIN–MADISON

2025

Date of final oral examination: 12th June 2025

The dissertation is approved by the following members of the Final Oral Committee:

Xuhui Huang, Professor, Chemistry (advisor)
Arun Yethiraj, Professor, Chemistry
Jeff Martell, Assistant Professor, Chemistry
Anthony Gitter, Associate Professor, Biostatistics and Medical Informatics

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my advisor Professor Xuhui Huang for his support, guidance, and encouragement throughout my doctoral journey.

I am also thankful to all current and former members of our lab for creating a collaborative and diverse research environment, including Dr. Siqin Cao, Dr. Peter Cheung, Dr. Ilona Christy Unarta, Dr. Jordy Homing Lam, Dr. Wei Wang, Dr. Xiaowei Wang, Dr. Yeqing Yu, Dr. Sam Chong, Dr. Cong Pan, Dr. Rick Xinzhou Xu, Dr. Yuqing Wang, Dr. Kirill Konovalov, Dr. Chu Li, Dr. Michael Suarez, Song Liu, Jimmy Ka Hei Choy, Jacqueline Cheryl Sabrina, Dr. Zhuo Liu, Dr. Yunrui Qiu, Andrew Yik, Eshani Goonetilleke, Bojun Liu, Michael O'Connor, Michael Kalin, Yue Wu, Zige Liu, Yichong Lao, Longbang Liu, Andres Lira, Peter C. Swanson, and Chengwei Dong.

I would especially like to acknowledge Dr. Jordy Homing Lam and Dr. Michael Alexander Suarez Vasquez, whose foundational work laid the groundwork for my own research. I am grateful to Dr. Congmin Yuan for sharing expertise in protein docking and visualization. I thank Ilona Christy Unarta for her guidance in setting up force fields and performing energy minimization. I appreciate Dr. Siqin Cao for engaging scientific discussions throughout our collaboration. I gained valuable insights and exposure to state-of-the-art techniques through collaboration with Bojun Liu, a highly responsible and dedicated collaborator. I also thank Zige Liu for valuable discussions, and most importantly for driving me to Costco.

I am fortunate to have had the opportunity to collaborate with individuals outside our group. I thank Dr. Junzhuo Liao and Professor Weiping Tang for their insight on fragment-based drug discovery, SARS-CoV-2, molecular glues and PROTACs systems, which greatly enriched this project. I am also honored to have collaborated with Ziru Chen, Yifan Li, Professor Huan Sun, and other outstanding researchers from diverse disciplines at The Ohio State University on LLM-agent–related projects.

I am sincerely grateful to my dissertation committee for their generosity with time and effort in attending my defense, reviewing my thesis, and offering thoughtful feedback. I thank Professor Arun Yethiraj for consistent guidance and thoughtful feedback across all stages of my research. I appreciate Professor Jeff Martell for the opportunity to engage in experimental collaborations and for the chance to learn deeply about protein engineering. I thank Professor Anthony Gitter for his instruction in a related course and for introducing me to the concepts and techniques of computational network biology, which deepened

my understanding of biomolecular interactions and regulation beyond the scope of proteins. In addition, his insight into explaining the probability density envelope in the work of FeatureDock significantly enhanced the conceptual clarity and interpretability of the method.

Finally, I wish to express my deepest gratitude to my boyfriend, Shuchen Yan, for his companionship. Home-cooked food, weekend board games, video games and our shared travels have brought me joy and cherished memories during this intense academic journey. A special thank also goes to my cat, Asparagus, for providing constant emotional support, being comforting and fuzzy.

Abstract

Proteins, owing to their structural complexity and functional diversity, play a pivotal role in drug development. Recent advances in deep learning have considerably expanded the capacity to analyze protein structures, infer physicochemical properties, and model intermolecular interactions. Nevertheless, the vastness of chemical space and the intricate nature of protein-ligand interactions pose significant challenges in the early stages of drug discovery.

This dissertation first addresses these challenges through the development and application of a comprehensive pipeline for fragment-based drug discovery (FBDD). A key contribution is the implementation of an automated data curation pipeline, including fetching, cleaning, and ligand decomposition tailored for ChemPLAN-Net, a deep learning framework designed to predict protein-fragment interactions using physicochemical features. This pipeline enables efficient training dataset generation and supports the systematic training ChemPLAN-Net across multiple protein families. Based on fragment predictions, virtual synthesis, screening and fragment linking are explored to develop fragment hits into candidate full ligands.

One critical challenge identified during this process is the accurate prediction of fragment poses and orientations within binding pockets, which significantly impacts downstream fragment elaboration. ChemPLAN-Net, despite its efficacy in fragment identification, lacks the ability to resolve fragment posing and fragment optimization (growing, linking, and merging). To address this limitation, we introduce FeatureDock, a transformer-based deep learning model that reframes the docking problem as a spatial probability density prediction task within a more coarsely grained space, by discretizing the protein surface and learning from physicochemical features of grid points. FeatureDock introduces an interpretable probability envelope over potential binding regions, based on which compound scoring and posing estimation is possible.

By integrating deep learning methodologies with protein structures and physicochemical features, this work contributes to the advancement of data-driven, interpretable, and cost-efficient drug discovery frameworks. The approaches developed herein aim to improve hit identification, reduce design-cycle latency, and ultimately increase the success rate of early-stage drug discovery pipelines.

List of Publications

1. Suarez Vasquez, M. A., Xue, M., Lam, J. H., Goonetilleke, E. C., Gao, X., Huang, X. (2021). "ChemPLAN-Net: A deep learning framework to find novel inhibitor fragments for proteins." *bioRxiv*, 2021–08. Cold Spring Harbor Laboratory.

2. Liu, B., Xue, M., Qiu, Y., Konovalov, K. A., O'Connor, M. S., Huang, X. (2023). "Graph-VAMPnets for uncovering slow collective variables of self-assembly dynamics." *The Journal of Chemical Physics*, 159(9). AIP Publishing.

3. Qiu, Y., O'Connor, M. S., Xue, M., Liu, B., Huang, X. (2023). "An efficient path classification algorithm based on variational autoencoder to identify metastable path channels for complex conformational changes." *Journal of Chemical Theory and Computation*, 19(14), 4728–4742. ACS Publications.

4. Xue, M., Liu, B., Cao, S., Huang, X. (2025). "FeatureDock for protein-ligand docking guided by physicochemical feature-based local environment learning using transformer." *npj Drug Discovery*, 2(1), 4. Nature Publishing Group UK London.

5. Liu, B., Cao, S., Boysen, J.G., et al. (2025). "Memory kernel minimization-based neural networks for discovering slow collective variables of biomolecular dynamics." *Nature Computational Science*. Nature Publishing Group UK London.

6. Chen, Z., Chen, S., Ning, Y., Zhang, Q., Wang, B., Yu, B., Li, Y., Liao, Z., Wei, C., Lu, Z., et al. (2024). "Scienceagentbench: Toward rigorous assessment of language agents for data-driven scientific discovery." *arXiv preprint arXiv:2410.05080*.

Contents

List of Tables

List of Figures

1 Computational Strategies and Challenges in Drug Discovery

1.1 Introduction

Computational methodologies[7] have emerged as transformative instruments in modern drug discovery, offering both speed and scalability to a field historically dominated by laborious and costly experimental screening. While high-throughput screening (HTS) technologies can now evaluate up to 10^6 compounds per day[8], this capacity remains dwarfed by the estimated size of drug-like chemical space, which exceeds 10^{60} molecules[9]. The vastness of this space renders exhaustive empirical screening fundamentally infeasible. Meanwhile, the development of a single clinically approved drug typically spans over a decade and incurs an average cost exceeding 2 billion US dollars, with the majority of candidates failing during preclinical or clinical stages due to issues related to efficacy, selectivity, pharmacokinetics, or toxicity[10]. This high attrition rate underscores the urgent need for more predictive and cost-efficient approaches in early-stage drug discovery.

In this context, computational approaches offer a powerful and cost-effective complement to experimental techniques. They enable researchers to virtually evaluate and prioritize candidate molecules at lower time and cost compared to physical assays. By simulating molecular interactions and dynamics (e.g., through molecular dynamics), predicting binding modes and affinities (e.g., via molecular docking), and learning patterns from quantitative structure–activity relationships (QSAR), *in silico* methods can pre-filter massive compound libraries, enrich hit rates, and guide experimental efforts toward the most promising leads. Computational methods intend to narrow the gap between *in silico* promise and *in vivo* performance.

The remainder of this chapter surveys how computational approaches are reshaping key challenges across the drug discovery pipeline, including chemical space exploration, structural biology, and deep learning methods intersecting with the modern drug discovery, laying the conceptual foundation for the work presented in this dissertation.

1.2 Role of protein structure and dynamics in drug design

Proteins serve as the primary molecular targets in pharmacotherapy due to their central involvement in virtually all cellular processes, including signal transduction, gene expression, metabolism, and immune responses[11]. Therapeutic interventions typically aim

to modulate aberrant protein functions arising from disease mutations, dysregulation, or pathogen interaction. The suitability of a protein as a drug target depends on multiple factors including its structural tractability, functional relevance, and accessibility to drug-like molecules.

Recently, advances in structural biology, bioinformatics, and chemical biology have dramatically expanded the druggable proteome, while also exposing the limitations of conventional small-molecule strategies.

1.2.1 Protein structures

A protein is generally considered druggable[12, 13] if it contains well-defined binding pockets capable of forming high-affinity, selective interactions with small molecules. Structure-based drug design (SBDD)[14], empowered by experimental approaches including X-ray crystallography, cryo-EM[15], and NMR, has been instrumental in targeting such proteins at atomic resolution.

Moreover, homology modeling[16] and AI-driven structure prediction (e.g., ESMFold[17] and AlphaFold[18]) have dramatically improved access to structural templates for proteins lacking experimental coordinates, facilitating broader application of SBDD across the human proteome[19]. Automated pocket detection algorithms like Fpocket[20] and DeepSite[21] quantify pocket volume, depth, and hydrophobicity to predict druggability across protein structures.

However, druggability is not solely a structural property. Many proteins undergo large-scale conformational rearrangements between active and inactive states or between open and closed forms[22]. These dynamics often give rise to cryptic binding sites which are transient pockets not evident in static structures but become druggable under certain conformational states[23]. Accounting for protein flexibility is essential not only to identify such cryptic pockets but also to recognize distal allosteric sites. Allosteric inhibitors offer enhanced selectivity by targeting regions that are less conserved across protein families, thus reducing off-target interactions. This is particularly valuable in highly conserved target classes like kinases and proteases[24].

Despite significant advances, a substantial portion of disease-relevant proteins remain "undruggable" by traditional small-molecule therapeutics[25]. These include transcription factors, scaffolding proteins, and RNA-binding proteins that lack classical pockets or exhibit flat, featureless surfaces, which often lack deep and well-defined pockets. Historically considered intractable, these proteins are often involved in oncogenesis, neurodegeneration, or infectious diseases, making them high-value targets for next-generation therapeutics.

Two breakthrough strategies, including PROTACs (proteolysis-targeting chimeras)[26] and molecular glues[1, 27], have redefined the concept of druggability by enabling targeted protein degradation rather than inhibition. PROTACs are bifunctional molecules that simultaneously bind the target protein and an E3 ubiquitin ligase, facilitating ubiquitin-mediated proteasomal degradation of the target. This approach bypasses the need for occupancy-based inhibition and allows modulation of previously inaccessible targets. Molecular glues are monovalent compounds that stabilize PPIs, promoting assemblies between target and E3 ligase that lead to degradation. Targeted degradation have yielded clinical candidates[28], such as BRD9 targeted degrader FHD-609[29], demonstrating the therapeutic viability of targeting undruggable proteins through induced proximity mechanisms.

1.2.2 Conformational changes

Conventional methods like X-ray crystallography and cryo-EM provide high-resolution snapshots, but they typically capture static snapshots and may overlook dynamic or low-populated conformations. Experimental techniques like single-molecule FRET[30] offer dynamic information but suffer from limited spatial resolution and scalability. Molecular dynamics (MD) simulations complement these limitations by simulating atomic movements over time.

MD simulations can uncover transient conformations, explore ligand entry and exit pathways, and reveal allosteric transitions inaccessible to experimental methods. These capabilities are especially valuable for investigating cryptic pockets[31] and allosteric regulation[32], which often require sampling rare conformational states. However, a key limitation of MD simulations is that they operate on femtosecond timescales, following Newtonian mechanics. As a result, they struggle to sample conformational changes that typically take place over microsecond to millisecond timescales, often failing to overcome large energy barriers and becoming trapped in metastable states.

Markov State Models (MSMs)[33, 34] were proposed as one of the methods to address the above mentioned challenges, by reconstructing long-timescale dynamics from ensembles of short trajectories. MSMs discretize the conformational space into states and estimate transition probabilities between them under a defined lag time. When applicable, MSMs enable prediction of protein folding[35] and allosteric pathways[36] beyond the duration of individual simulations.

While effective, MSMs are built on the Markovian assumption, which may not always hold especially in complex biomolecular systems with long memory. To overcome

this limitation, non-Markovian frameworks that incorporate memory effects have been proposed[37]. A recent advancement in this area is Memory kErnel Minimization neural networks (MEMnets)[38], which incorporate the integrative generalized master equation (IGME) theory into a deep learning framework. MEMnets identify optimal collective variables (CVs) that capture the essential slow dynamics, by training weight-shared neural networks across multiple time-lagged frames to project high-dimensional MD data into a low-dimensional latent space where memory kernels are minimized. For instance, this method discovered two parallel folding pathways in WW-domain, outperforming its Markovian counterparts such as time-lagged independent component analysis (tICA)[39] and VAMPNets[40].

All these approaches are working toward the goal of accurately modeling the dynamic conformational landscape of proteins. By better understanding and predicting protein dynamics, we improve our ability to find and optimize ligands especially for challenging cases such as cryptic pockets and unconventional targets.

1.3 Overview of early drug discovery stages

The driving hypothesis for druggable proteins is that the specific small molecule can be found for a given disease-relevant protein target, which geometrically fits in the binding pocket and terminates pathogenic mechanism by proper physicochemical interactions without causing fatal harm to the living organisms[41]. The realization of this goal involves a multistage, resource-intensive pipeline, traditionally encompassing target identification and validation, hit discovery, hit-to-lead optimization, and preclinical candidate selection[42].

Over the last decade, the practical implementation of these stages has evolved dramatically owing to increased automation, including the ultra-high-throughput experimentation[43] and the wide use of artificial intelligence[44, 45]. This section reviews each stages before clinical trials in detail, emphasizing how computational methodologies help reduce cycle time, enhance success rates and expand the accessible chemical design space.

1.3.1 Target identification and validation

Target identification[46, 47] involves selecting a specific biological molecule or pathway that modulates a particular disease. This selection is increasingly informed by "-omics" data (genomics, transcriptomics, proteomics, metabolomics)[48] that can reveal critical disease drivers. Computational approaches[49] play a vital role in this stage by analyzing large-scale biological datasets to prioritize potential targets. For instance, network biol-

ogy based on genome-wide association studies (GWAS)[50] and quantitative trait loci (QTL)[51] can identify key nodes in disease pathways[52]. Deep learning models like deepDTnet[53] identified novel drug-repurposing targets from a heterogeneous drug—gene–disease network.

Target validation[54, 55, 56] rigorously confirms the causal relationship between the selected protein and the disease phenotype. Experimental validation often involves genetic manipulations (e.g., knockout, knockdown, overexpression), small-molecule tool compounds, expression profiling, and phenotypic assays. Computational tools support this by simulating target modulation in biological networks, and more recently by interpreting target-disease association data[57].

1.3.2 Hit identification

Hit discovery remains the principal funnel that converts an expansive chemical universe into tractable lead compounds for downstream optimization. Three primary methodologies play increasingly important roles at this stage through automation, including traditional high-throughput screening (HTS), structure- or ligand-based virtual screening (VS) and fragment-based drug discovery (FBDD).

1.3.2.1 High-throughput screening (HTS)

High-throughput screening (HTS) is a mature yet actively developing experimental technique to find small molecules with satisfying activities, by screening an exhaustive compound database over the interesting target[58].

Robotic HTS platforms routinely execute 384- and 1,536-well biochemical or cell-based assays at throughputs approaching $10^5 - 10^6$ wells per day[59]. However, this throughput still samples only a small fraction of the commercially accessible small-molecule space[41]. In addition, HTS remains costly and is often constrained by the complexity of assay development, cell model robustness, and compound stability. These limitations have catalyzed the integration of HTS with computational methods, including virtual screening and fragment-based approaches, to enhance discovery efficiency.

1.3.2.2 Virtual screening

Among physics-based methods, MD simulations combined with free energy methods stand out for providing quantitative accuracy. In particular, Free Energy Perturbation (FEP)[60, 61, 62] has been a rigorous method in drug discovery. This method computes the change in binding free energy between two related ligands $\Delta\Delta G_{AB}$ by transforming

one ligand into the other alchemically in the protein's binding site via MD simulation. Despite its computational demands, modern FEP implementations such as FEP+[61] have achieved remarkable accuracy within 1 kcal/mol of experimental values, making them highly predictive especially in lead optimization settings.

Structure-based virtual screening (SBVS) by molecular docking[63] is a more cost-efficient approach of structure-based drug discovery in the early hit-finding stage, accessible for screening up to billions of compounds[64]. Docking programs sample possible binding poses for each molecule and score them according to shape complementarity, hydrogen bonding, hydrophobic contacts, and other simplified physicochemical criteria. The outcome is a ranked list of candidate ligands predicted to bind the protein. Top-ranked compounds are then tested experimentally to confirm activity. SBVS historically relied on rigid docking and physics-inspired scoring, but its convergence with machine learning has shifted the performance frontier. The design of scoring functions, traditional docking methods and deep learning-based docking methods will be discussed in detail in Chapter 4 Section 4.2.

Ligand-based virtual screening (LBVS) [65] is employed when the target protein is unknown or when SBVS is impractical. LBVS relies on the principle that structurally similar molecules tend to exhibit similar biological activities. This approach utilizes information from known active compounds to identify new candidates with potential bioactivity. Traditional LBVS techniques include similarity searching[66], pharmacophore modeling[67, 68] and QSAR modeling[69]. Traditional QSAR models using techniques like Naïve Bayesian models or support vector machines were used to predict biological activity of molecules[70]. In recent years, deep learning models have largely overtaken classical QSAR modeling due to their ability to automatically learn complex features from large datasets. They are also revolutionizing de novo drug design. These applications will be discussed in detail in Chapter 2, together with molecular representations and deep learning architectures.

1.3.2.3 Fragment-based approaches

Fragment-Based Drug Discovery (FBDD)[71] has emerged as a powerful alternative and complement which efficiently reduces the chemical space to search. The core principle of FBDD is that small, low-complexity molecules (typically with molecular weights $<$ 300 Da, often $<$ 250 Da, adhering to the "Rule of Three"[72]) can bind to distinct sub-pockets on a protein target with higher efficiency and probability than larger, more complex drug-like molecules. Although these initial interactions are often weak (micromolar to millimolar affinity), the identified fragments serve as high-quality starting points for chemical elaboration into more potent, lead-like compounds. This approach offers a key advantage in enabling more efficient sampling of chemical space due to the smaller size

and reduced complexity of the library components.

The FBDD workflow typically begins with the a curated library of diverse and representitive fragments, followed by a fragment hit identification. Experimental hit identification techniques include sensitive biophysical techniques capable of detecting weak binding, such as differential scanning fluorimetry (DSF), nuclear magnetic resonance (NMR), surface plasmon resonance (SPR), isothermal titration calorimetry (ITC), and X-ray crystallography[73].

Computational methods play an increasingly sophisticated role. Beyond designing diverse and pharmacophorically rich fragment libraries, virtual fragment screening (VFS) can prioritize fragments for experimental testing. Recently, deep learning approaches like ChemPlan-Net[4] have been developed to predict potential inhibitor fragments by learning from the physicochemical features of known protein-ligand complexes, offering an alternative to, or augmentation of, traditional docking-based fragment screening. Such methods aim to more accurately identify fragments that are likely to bind productively to specific target sites.

Once fragment hits are identified and their binding modes validated from the above methods, the critical step of evolving these low-affinity hits into potent leads begins. This can be achieved through several strategies[74, 75]:

- **Fragment Growing**: Extending a single fragment by adding new chemical functionalities that make additional favorable interactions within an adjacent sub-pocket. Computational approaches, including structure-based design and analysis of protein-ligand interaction fields (e.g., AutoGrow[76]), guide the design of these additions. Modern deep learning models are also being applied to fragment growing, where generative approaches such as SRIFE[77] can suggest novel chemical extensions in the context of the protein binding site, aiming to optimize interactions and properties.

- **Fragment Linking**: Connecting two or more fragments that bind to distinct, often adjacent, sites on the target. This requires careful design of a linker that maintains the optimal binding orientation of each fragment and possesses favorable physicochemical properties. Traditionally[78], this involves searching linker databases or manual design. However, deep generative models have revolutionized this area. For instance, graph-based models and structure-aware like DeLinker[79] can design novel linkers by considering the 3D structural context of the bound fragments. These methods enable the exploration of a much broader and more novel chemical space for linkers than previously feasible.

- **Fragment Merging**: Combining two overlapping fragments that bind to adjacent or partially overlapping sites on the target protein into a single, more potent compound. This strategy involves identifying shared pharmacophoric features and designing a merged molecule that retains critical interactions from both parent fragments. Recent advancements have utilized graph databases and computational algorithms to sample diverse chemical merges, enhancing the efficiency and success rate of this approach[80].

Among these three strategies[81], fragment growing via chemical synthesis is the most widely used in practice. Fragment linking, while less commonly used, is more efficient at achieving substantial gains in binding affinity by simultaneously engaging multiple sub-pockets. Fragment merging, by contrast, remains the least utilized approach, but is gaining more attention with the evolution of generative models[75].

1.3.3 Hit-to-lead optimization

During this resource-intensive hit-to-lead (H2L) stage[82, 83] of pre-clinical discovery, promising hit compounds identified from the screening phase are iteratively modified and evaluated to enhance multiple critical properties simultaneously. The goal is to develop lead compounds that possess not only high target affinity and efficacy but also favorable ADMET (absorption, distribution, metabolism, excretion, and toxicity) properties, appropriate selectivity against off-targets, and suitable physicochemical characteristics (e.g., solubility, permeability, stability) for *in vivo* administration and eventual drug development. Achieving an optimal balance among these properties is essential for the progression of compounds into clinical development.

Computational methods mentioned all above play a pivotal role in guiding H2L optimization, including MD simulations, FEP calculations, molecular docking, and QSAR modeling. FEP, in particular, has gained prominence at this phase for its ability to provide accurate estimations of relative binding free energies, for limited number of surviving hits from the screening stage. Furthermore, techniques like scaffold hopping or bioisosteric replacement[84], often aided by computational analysis of shape and electrostatic similarity, are employed to navigate away from problematic chemotypes or intellectual property challenges while retaining desired activity.

The iterative DMTA (design, make, test analyze) cycle[85, 86] is paramount in lead optimization. Computational predictions inform synthetic priorities, experimental data from newly synthesized compounds feeds back into model refinement, and this loop continues until candidate molecules meeting stringent preclinical criteria are identified. The

integration of deep learning and automated synthesis platforms promises to accelerate this complex phase. For example, AI-driven molecule generation conditioned on specific molecular properties has been enabled by generative models trained on large chemical datasets [87]. In addition to advances in supervised models and generative models, reinforcement learning (RL) has gained attention as a promising strategy for molecular optimization Frameworks such as ReLeaSE (Reinforcement Learning for Structural Evolution)[88] and GENTRL[89] have demonstrated the ability to propose novel molecules with desired properties.

1.4 Thesis Overview

- **Chapter 2**. This chapter introduces foundational concepts linking molecular representations to neural network architectures by systematically reviewing various forms of molecular encoding and neural network architectures. The chapter also highlights my coauthored works (FeatureDock, GraphVAMPNet, ChemPLAN-Net, etc) as practical demonstrations of how deep architectures can extract sequential, spatial, and dynamical features from chemical and biological data.

- **Chapter 3**. This chapter details the dataset curation, architecture design and training strategy of ChemPLAN-Net, a CNN-based model that predicts inhibitor fragments based on physicochemical environments around protein binding sites. ChemPLAN-Net illustrates how fragment-based strategies can benefit from localized spatial learning and curated fragment libraries. This chapter further includes fragment-based methods to apply ChemPLAN-Net predicted binding fragments to entire candidate ligands, including virtual synthesis and LBVS screening, and deep learning based fragment linking using DeLinker.

- **Chapter 4**. This chapter presents FeatureDock, a novel deep learning model for structure-based docking. FeatureDock reformulates the docking problem as a point-wise prediction task, using a Transformer encoder to process spatially embedded physicochemical features at 3D grid points in the protein binding pocket. This design allows the model to learn long-range spatial dependencies and infer ligand-compatible regions on protein surface. FeatureDock demonstrates not only competitive performance but also enhanced interpretability through its attention-based architecture. The model represents a step forward in integrating deep learning with spatial docking workflows and serves as a proof of concept for future Transformer-based biophysical models.

- **Chapter 5**. The chapter concludes the findings presented in this dissertation and outlines key challenges and opportunities for future research.

2 Application of Deep Learning to Chemical Data

Deep learning has shown great promise for chemical and biological data analysis, enabling more accurate prediction and accelerating tasks in drug discovery. Neural network models can capture complex and nonlinear patterns from diverse inputs, complementing traditional rule-based and physics-based methods. Especially in recent years, the availability of massive chemical datasets (e.g. $> 10^8$ compounds in PubChem[90, 91, 92, 93], ZINC[94, 95, 96, 97] and ChEMBL[98, 99, 100, 101]) has enabled neural networks to tackle challenges from data at unprecedented scale. This has dramatically accelerated *in silico* pipelines.

Small molecules[102, 103] and proteins[104, 105, 106] can be represented in multiple formats suitable for different deep learning architectures. Common representations include numerical vectors of fixed-length (e.g. molecular descriptors, fingerprints) that serve as input to multi-layer perceptrons (MLPs), linear notations (e.g. SMILES, FASTA) which are often processed by sequence-based models (e.g. RNNs or Transformers), graph-based encodings where atoms are nodes and bonds are edges for graph neural networks, and 3D molecular representations (e.g. voxels, cloud points) for 3D convolutional neural networks and geometric deep learning models.

The remaining of this chapter will discuss the representations (in Section 2.1), neural network architectures (in Section 2.2) and their applications in detail. Where applicable, the discussion will highlight contributions from my own co-authored works, demonstrating how different representations and models are instantiated in state-of-the-art research:

- ChemPLAN-Net[4]: A CNN-based framework for predicting fragment inhibitors.
- GraphVAMPNet[107]: An extension of SchNet to study long-timescale dynamics of self-assembling patchy particles.
- Latent-space path clustering (LPC)[108]: An unsupervised method leveraging VAE latent representations for clustering kinetic transition pathways.
- FeatureDock[2]: A Transformer-based approach for protein-ligand docking using physicochemical features of grid points.
- MEMNets[38]: An architecture designed to learn non-Markovian slow dynamics from MD trajectories.
- ScienceAgentBench[3]: A benchmarking framework for evaluating the performance of large language model (LLM)-based coding agents in scientific discovery tasks.

2.1 Molecular representations

2.1.1 Molecular descriptors and fingerprints

2.1.1.1 Molecular descriptors

One of the earliest and most influential frameworks in drug discovery is Lipinski's Rule of Five (Ro5)[109], where simple physicochemical criteria are used to evaluate the oral bioavailability of drug-like molecules, including molecular weight $\leqslant$ 500 Da, no more than five hydrogen bond donors, no more than ten hydrogen bond acceptors, and logP $\leqslant$ 5. This approach exemplifies the broader concept of using a "bag of properties", where each molecule is characterized by a vector of physicochemical, topological, and electronic properties. Descriptor toolkits such as PaDEL-descriptor[110] and the Chemistry Development Kit (CDK)[111, 112], automate the extraction of hundreds of descriptors, making them attractive and valuable for modeling.

2.1.1.2 Fingerprints

In parallel to descriptors, molecular fingerprints were developed to encode molecular structure in a format optimized for similarity searching and rapid database screening. Fingerprints represent molecules as bit strings or count vectors, indicating the presence or frequency of a predefined substructure or local chemical environment. Two widely used fingerprint types include:

- **MACCS (Molecular ACCess System) keys[113]**. It consists of 166 predefined atom-based and subgroup 2D descriptors.
- **Extended Connectivity Fingerprints (ECFPs)[114]**. It is also known as Morgan fingerprints and capture atom-centered circular neighborhoods to encode local chemical environments up to a given radius (Figure 2.1).

These representations are widely used in large-scale chemical databases such as PubChem, where they serve as the basis for similarity and substructure searching[115]. The computational efficiency of these fingerprints makes them particularly well-suited for handling the massive chemical space in public repositories at the tasks of compound clustering, diversity analysis, and virtual screening.

Beyond ligand-centric modeling, descriptors have found applications in protein-ligand[116, 117] and protein-protein[118] prediction, often through residue-level and interaction-based descriptors. In the context of molecular dynamics (MD), fingerprints derived from time-resolved trajectories have been used to characterize protein flexibility, identify relavant

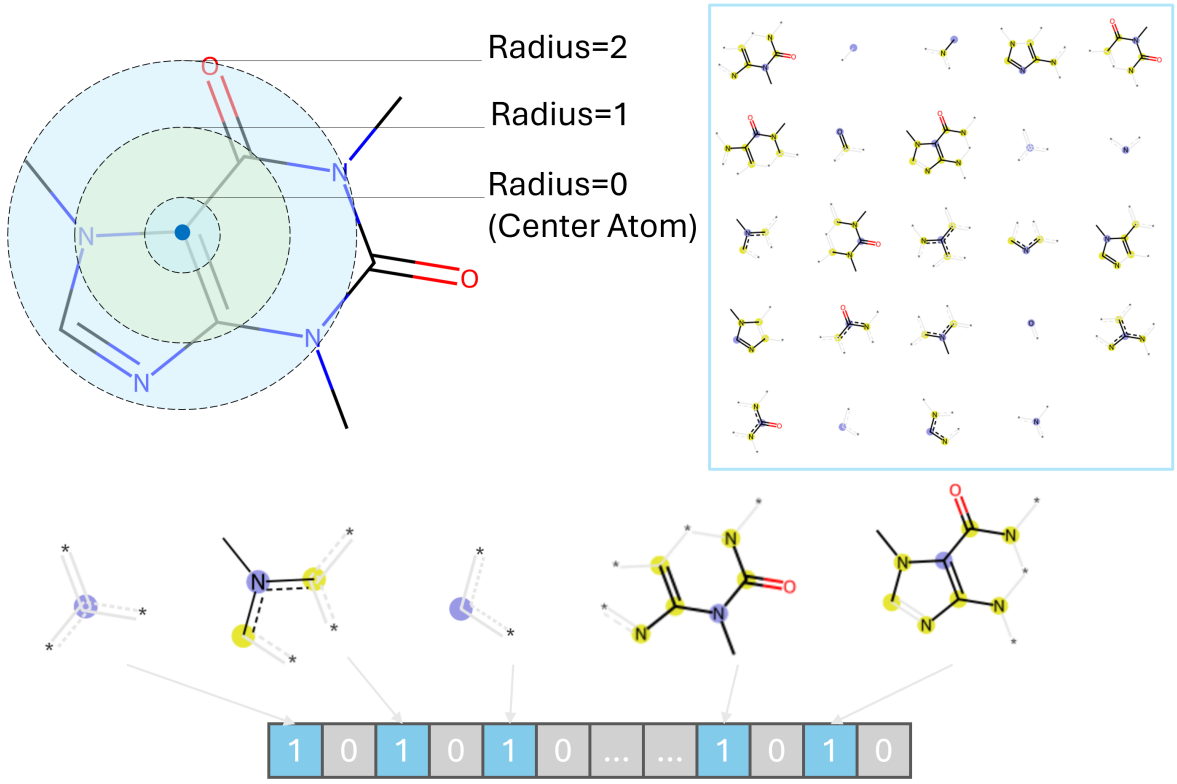


Figure 2.1: Example of an ECFP4 (Morgan) fingerprint for caffeine. In ECFP4, atom neighborhoods are hashed to a radius of 2 (highlighted by concentric shells), producing the bit vector that encodes local environments of the molecule. Subgraphs are listed in the blue box. Each subgraph hash is then mapped to one position in a fixed-length binary vector, setting that bit to 1.

protein conformation, which can further be used in virtual screening[119] and solvation free energy prediction[120].

2.1.1.3 Strengths and limitations

Molecular descriptors and fingerprints predate the modern deep learning era and play a significant role in quantitative structure-activity relationship (QSAR) modeling[121, 122]. Engineered features remain computationally inexpensive, reproducible and chemically interpretable. The representation of a fixed-length numerical or binary vector, is compatible with classical machine learning methods, including linear regression, decision trees, support vector machines (SVMs), or shallow neural networks.

However, these handcrafted representations exhibit several critical limitations. De-

scriptors, especially those based on 2D topology, often fail to capture stereochemistry, chirality, and conformational flexibility that are crucial for molecular recognition and activity in biological systems[123]. Fingerprints, particularly those based on hashing such as ECFPs, may suffer from bit collisions[124], where distinct substructures map to the same index, thereby reducing resolution and specificity. Additionally, feature engineering demands domain expertise and may introduce biases if important features are omitted or overrepresented[125]. These limitations have motivated the shift toward representation learning methods that aim to extract task-specific features directly from raw molecular formats such as SMILES strings, molecular graphs, and 3D atomic coordinates.

2.1.2 String-based representation

String-based representations provide a compact and human-readable format to encode molecular structures as linear sequences of characters. These representations have been foundational in cheminformatics, especially in machine learning due to their simplicity, efficiency, and compatibility with sequence processing models such as recurrent neural networks (RNNs) and Transformers. By transforming molecules into linear sequences, these encodings facilitate the application of natural language processing (NLP) techniques to chemical data.

2.1.2.1 Small molecules

The most widely used molecular string format is the Simplified Molecular Input Line Entry System (SMILES)[126, 127]. SMILES strings describe molecular graphs using ASCII characters, encoding atoms, bonds, branching, ring closures, and aromaticity. It operates via a depth-first traversal of the molecular graph, producing a compact yet expressive encoding, as summerized in Table 2.1. Isomeric SMILES extend this format by including stereochemical and isotopic information, which is essential for encoding chiral centers and cis-trans isomerism. The SMILES Arbitrary Target Specification (SMARTS) language generalizes SMILES with wildcard atoms, recursive environments and logical operators, enabling precise sub-structure search in RDKit[128] and other similar toolkits.

Because different traversal orders can lead to different SMILES strings for the same molecule (Figure 2.2), canonicalization algorithms have been developed to create a one-to-one mapping between SMILES strings and molecules. Canonicalization algorithms such as CANGEN[127], Morgan[129], RDkit implementation[130] and the universal SMILES[131] implement distinct heuristics to generate unique canonical forms. However, because these heuristics differ, canonical SMILES may vary across algorithms.

Table 2.1: Daylight SMILES grammar with extensions to capture stereochemistry, isotopes, and SMARTS patterns.

Pattern	Encodes	Examples	Notes
`B, C, N, O, P, S, F, Cl, Br, I`	Organic subset atoms (no brackets)	`CCO` (ethanol)	valence and implicit H are inferred.
`b, c, n, o, p, s` (lower-case)	Aromatic atoms	`c1ccccc1` (benzene)	Allowed only inside aromatic rings.
`-` (or omitted)	Single bond	`CC`	
`=, #, :`	Double, triple, aromatic bonds	`C=O`, `C#N`, `c1:c:c:c:c:c1`	`:` can be omitted
`( )`	Branching	`CC(=O)O` (acetic acid)	
Digits `1-9`, `%10`	Ring closures	`C1CCCCC1` (cyclohexane)	First digit opens ring, second closes; use `%` for numbers $\geqslant 10$.
`.`	Disconnected fragments	`Na.[O-]C(=O)C` (sodium acetate)	
`*`	Wildcard atom	`C(*)C`	Matches any atom (SMARTS context)
`>`	Reaction SMILES separators	`reactants > agents > products`	Middle "agents" block is optional.
`:n`	Atom–mapping index (reactions)	`[C:1]=[O:2]`	Tracks atom correspondence across sides.
`[<mass>]`	Isomeric prefix	`[2H]O` (D_2O); `[13C]C`	
`/` *or* `\`	E/Z double-bond stereochemistry	`F/C=C\F` (cis-1,2-difluoroethene)	
`@`, `@@`	Tetrahedral chirality (R/S)	`F[C@@](Cl)(Br)I`	`@` = clockwise, `@@` = anticlockwise

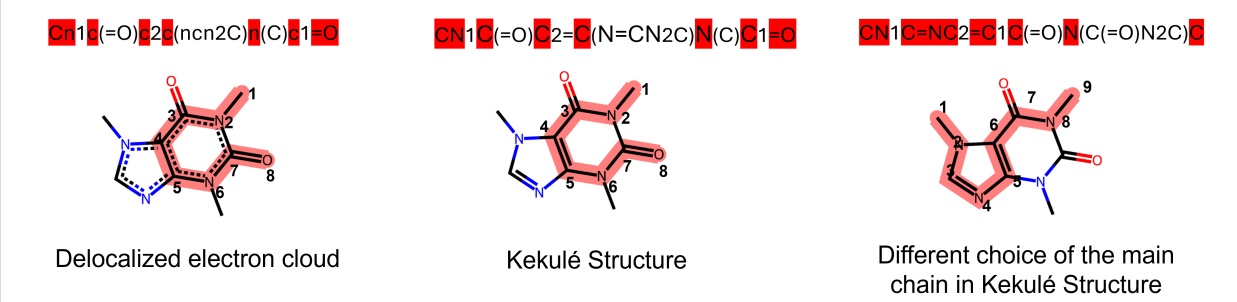


Figure 2.2: Three valid SMILES representations of caffeine using different traversal and aromaticity conventions. Although graphically identical, the linear encodings differ, highlighting the need for canonicalization. **Left**: Delocalized form uses aromatic dotted bonds. **Middle**: Kekulé form places alternating single/double bonds. **Right**: An alternative main-chain traversal of the Kekulé form which follows a different linear walk through the graph. In every case the atoms that belong to the chosen SMILES backbone are highlighted in red, illustrating how a different way of depicting aromaticity or a different traversal describes the identical molecular graph.

SMILES has been widely adopted for machine learning applications, particularly in molecular property prediction[132, 133] and generation[134]. In generative chemistry, SMILES-based variational autoencoders (VAEs)[135, 136] have been employed in de novo molecular design, with reinforcement learning[137, 89] guided generation toward desired chemical properties.

However, a limitation of SMILES for generative models is its syntactic fragility that small character perturbations often result in invalid or chemically nonsensical molecules. To address this, alternative formats have been proposed. DeepSMILES[138] modifies the SMILES grammar to solve ring closure pairing and unbalanced branch parentheses for learning models, improving training stability and decoding fidelity. A more robust encoding is SELFIES (SELF-referencing Embedded Strings)[139], which guarantees that every syntactically valid SELFIES string corresponds to a valid molecule. This property is achieved through a self-referential grammar that enforces chemical valency constraints during string generation. SELFIES-based RNN models therefore achieve 100% validity without explicit grammar masks. Whereas DeepSMILES and SELFIES mitigate syntactic errors by redesigning the string grammar, an orthogonal strategy retains vanilla SMILES but modifies the learning paradigm. For example, reinforcement learning (RL) fine-tuning via REINVENT improved the validity of SMILES by penalizing syntactically incorrect outputs[140].

Other string-based representations for small molecules also exist. The IUPAC Interac-

tional Chemical Identifier (InChI)[141] provides a standardized and canonical representation of chemical substances. Unlike SMILES which is primarily designed for readability and molecular graph reconstruction, InChI provides a strictly-unique and canonical identifier that encodes connectivity, tautomerism, isotopes, stereochemistry, making InChI suitable for database indexing and cross-referencing. InChIKeys[142] are fixed-length, hashed versions of InChIs designed for rapid database searching. They are optimized for high-throughput web search and deployed by molecular repositories to link records, deduplicate entries and integrate external annotations. InChI and InChIKey are not typically used as model inputs, instead they serve a critical role in molecular identity management, retrieval, and interoperability across large-scale datasets.

A further design choice concerns tokenization, which maps the raw string into discrete symbols consumed by the RNNs or Transformers. The simplest approach, termed as character-level tokenizer, which treats each ASCII character as an independent token. However this is not commonly-used because it does not respect the SMILES grammar of multicharacter atoms such as "Cl" or "Br". One common choice is to use atom-aware tokenizers which split according to syntactic elements including atoms, ring digits, branch parentheses and bond symbols, forming the default in DeepChem's SmilesTokenizer[143]. This Regex-based atom-wise scheme balances interpretability and moderate sequence length. Likewise, Molecular Transformer reaction model[144] and its successor Chemformer[145] used another Regex-based tokenization method proposed specifically for reactions[146]. In order to compress recurring substructures, sub-word algorithms[147] such as Byte-Pair Encoding (BPE) have been adapted in ChemBERTa[148] to tokenize frequent chemical motifs in an end-to-end learnable manner.

2.1.2.2 Biological macromolecules

In addition to small molecules, string-based encodings have been extended to handle macromolecules such as polymers, peptides, proteins, and nucleic acids. A specialized form of SMILES, BigSMILES[149], has been developed to encode polymers by using repeating units and polymeric specific uncertainty.

For natural biopolymers, the primary sequence acts as a native string-based representation, which encodes amino acid or nucleic acid sequences in a linear format such as FASTA. This representation serves as the input for protein language models (PLMs) using transformer-based architectures trained on millions of sequences to learn biologically meaningful representations without requiring explicit structural information. The Evolutionary Scale Modeling (ESM) family of PLMs exemplifies this paradigm. ESM-1b[150] and ESM-2[17] demonstrated that self-supervised training on large-scale protein sequences yields

embeddings useful for mutational effect prediction, remote homology detection, and contact map estimation. The most recent iteration, ESM-3, introduces structured embeddings that integrate atomic-level resolution with sequence-based learning, narrowing the gap between language-only and structure-aware models[151]. Concurrently, AlphaFold2[152] revolutionized protein structure prediction by combining sequence-based attention, multiple sequence alignments (MSAs), and structural template information to predict protein structures with near-experimental accuracy. Building on this, AlphaFold3[153] extends capabilities beyond monomeric protein folding to encompass protein-ligand, protein-RNA, and protein-complex modeling, by incorporating diffusion-based generative modeling from sequence embeddings.

2.1.2.3 Strengths and limitations

String-based representations excel in simplicity, storage efficiency and compatibility with sequence models, making them ideal for massive libraries and large-scale parallel training. Especially, the use of protein sequences in PLMs like ESM and in structure predictors like AlphaFold demonstrates the remarkable expressiveness of string-based representations. These qualities have made string-based representations foundational to deep learning pipelines for small molecules and proteins.

However, these formats are not without limitations. As discussed in Section 2.1.2.1, SMILES strings can be syntactically fragile, and lack inherent 3D information. Errors in generation often produce invalid molecules, requiring additional postprocessing or filtering. Furthermore, string representations encode expressivity but omit explicit 3D geometry, limiting their expressiveness for tasks involving stereoelectronic effects or conformational dynamics. Hybrid architectures have emerged by combining SMILES with graph or 3D representations[154].

Additionally, modeling multiple protein conformations remains challenging. Recent approaches have been proposed to address protein flexibility modeling. For example, MSA clustering[155] has been used to bias AlphaFold2 toward alternative conformational states[156]. Other emerging techniques include combining AlphaFold with Monte Carlo tree search[157], or using latent diffusion models[158]. These strategies reflect the growing synergy between sequence-based models and physically grounded ensemble generation.

2.1.3 Graphs and geometry

Graph-based representations provide a natural and expressive formalism for modeling molecular structure. A molecule can be conceptualized as a graph, where atoms are

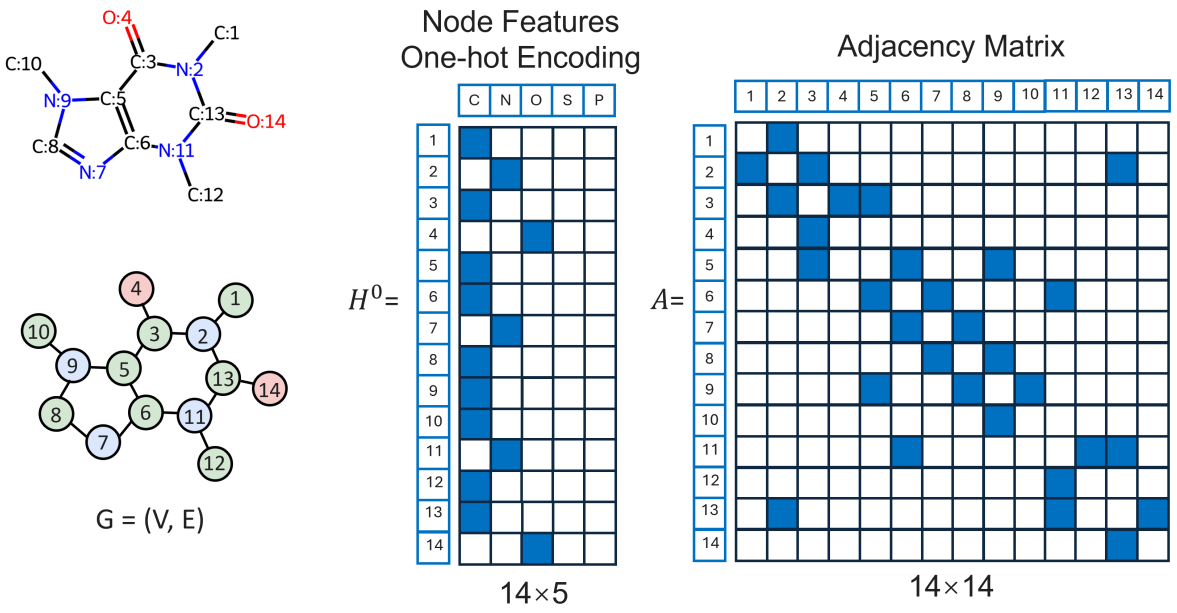


Figure 2.3: Constructing a machine-readable graph representation of caffeine. Each atom is featurized using a Onehot Embedding that encodes five element types: C, N, O, S, P, resulting in the initial node-feature matrix $\mathbf{H}^0$. Bond connectivity is captured by the adjacency matrix $\mathbf{A}$, where $\mathbf{A}_{ij}$ denotes whether an edge exists between the corresponding atom pair.

represented as nodes and chemical bonds as edges (Figure 2.3). This abstraction preserves chemical connectivity, remains invariant to atom ordering, and interfaces seamlessly with graph neural networks (GNNs) and SE(3)-equivariant models.

2.1.3.1 Structure files

Before describing graph representations in detail, it is important to acknowledge the role of the standardized file formats from which atomic coordinates and bond topologies are derived. Small molecule coordinates are commonly stored in SDF and MOL2 files, which record atom types, bond orders and optionally 3D conformers. Macromolecular structures are stored in PDB or mmCIF format. Public repositories such as PubChem, ChEMBL and ZINC supply millions of energetically minimized conformers of small molecules, whereas the RCSB PDB[159] and AlphaFold DB[18] provide experimental and predicted protein structures. These data sets underpin the graph-based inputs used in deep learning pipelines.

2.1.3.2 Molecular graphs

In molecular graph representations (Figure 2.3), each atom is associated with a set of features such as element type, formal charge, hybridization state, degree, and aromaticity. Similarly, each bond is annotated with features such as type (e.g., single, double, aromatic), stereochemistry (e.g., cis/trans), and ring membership. These categorical properties are typically one-hot encoded, while continuous features (e.g., partial charges) are normalized and embedded. The connectivity of atoms is represented as adjacency matrix or edge lists for sparse graphs to achieve storage efficiency. Graph types vary by their geometric resolution[160]:

- **2D graphs** encode only the adjacency matrix, omitting coordinates entirely.
- **2.5D graphs** enrich the 2D backbone with geometric scalars such as inter-atomic distances and bond angles.
- **3D graphs** present the atomic cartesian coordinates and include spatial information such as dihedral angles.

Graph representations are foundational to GNNs, which aggregates information from its neighboring nodes and edges, learning contextualized embeddings that reflect the local chemical environment. Through stacking multiple layers, these models learn to capture long-range dependencies. This framework has been widely adopted in molecular property prediction, such as solubility, toxicity[161], and quantum property prediction[162]. Beyond small-molecule property prediction, graph-based models have also been applied to protein-ligand binding affinity estimation, where both ligands and receptor environments are encoded as graphs. Using GNNs for both ligands and protein pockets[163] led to improved generalization compared to fixed descriptors, highlighting the expressiveness of learned structural features in bioactivity modeling.

In the domain of molecular dynamics, GraphVAMPNets[164, 107] combined graph-based learning models with variational approach for Markovian process (VAMP) theory to learn slow dynamics. This facilitates construction of Markov state models (MSMs)[33, 34] and conformational ensembles from simulation data .

For generative modeling, graph-based variational autoencoders (VAEs) and generative adversarial networks (GANs) such as GraphVAE[165] and MolGAN[166] have been developed to learn probability distributions over molecular graphs, enabling *de novo* generation of chemically valid compounds.

Building upon atomic graphs and geometric models, recent work has introduced hierarchical graph representations[167, 168] to better model molecules with complex, multi-resolution structures. In a hierarchical graph, a molecule is represented not only at the atom-bond level but also at higher levels of abstraction, such as chemical motifs. These coarser nodes are connected by edges reflecting chemical or spatial relationships at higher structural levels. Several deep learning models have operationalized this idea. The Hierarchical Graph Network (HierG2G)[167] constructs molecules from motifs using a multi-stage message passing scheme that first reasons over coarse substructures and then refines atom-level embeddings. However, challenges remain in defining optimal hierarchies, handling graph coarsening dynamically, and balancing information flow across levels. Nevertheless, hierarchical graph representations represent a promising frontier in molecular machine learning by combining the expressiveness of graphs with the structure-awareness of human chemical intuition.

2.1.3.3 Strengths and limitations

A central benefit of graph-based molecular representations is their invariance to node permutations, avoiding issues encountered in SMILES. Moreover, graphs naturally encode local chemical environments and extended topologies. However, graph-based approaches still face limitations. Unless explicitly integrated in node or edge features via feature engineering, basic graph representations omit 3D structural information.

2.1.4 Spatial points

In addition to graph and string-based representations, spatial molecular representations provide rich information about geometric and physicochemical information in three-dimensional (3D) space. The two most widely used spatial representations are voxel grids and point clouds. While voxel grids impose a structured lattice over molecular space, point clouds offer a grid-free, permutation-invariant alternative that has gained traction in recent deep learning applications. These formats are particularly valuable for modeling the structure and interactions of molecules in a spatially explicit manner, and have enabled significant advances in deep learning applications.

2.1.4.1 Voxel grids

Voxel-based representations discretize 3D space into a regular Cartesian lattice, where each voxel (e.g. a volumetric pixel of 1Å^3) or point contains scalar or vector-valued features describing local chemical properties. Each voxel typically stores values such as atomic

density, partial charge, and hydrophobicity. These features may be derived from a Gaussian smearing of atomic positions or other kernel-based smoothing techniques.

Voxel grids are particularly well suited for 3D Convolutional Neural Networks (3D CNNs), which scan the spatial volume with learnable filters to identify translationally invariant patterns of molecular shape and chemistry. This representation has been applied effectively in drug discovery for binding affinity prediction and ligand pose scoring, as in AtomNet[169] and KDEEP[170]. However, because standard 3D CNNs are not inherently rotationally invariant, extensive data augmentation is typically required during training to learn rotational invariance.

NucleicNet[171] and ChemPLAN-Net[4] employ convolutional neural networks (1D CNNs or 2D CNNs) to extract patterns from voxel-embedded physicochemical descriptors per point, learning protein-RNA or protein-fragment binding preferences. While these two frameworks do not capture spatial correlations across voxels as comprehensively as 3D CNNs, they offer computational efficiency as avoided the intensive data augmentation process. These approaches, however, could be extended to capture richer geometric context by incorporating equivariant neural networks that respect spatial symmetries. Motivated by the above two frameworks, FeatureDock[2] processes voxelized binding pocket features using a Transformer architecture. The attention-based mechanisms used in Transformer encoder achieved better predictive power and explainability, compared to CNNs of the same capacity.

2.1.4.2 Point clouds

Point clouds represent molecules or molecular environments as unordered sets of 3D coordinates, optionally augmented with atom-level or spatial features (e.g., element type, electrostatic potential, local curvature)[172]. Unlike voxel grids, point clouds are sparse and memory-efficient[173], avoiding issues related to discretization artifacts or grid alignment.

Recently, point clouds have gained significant attention in molecular generative modeling, especially for generative tasks involving flexible or large molecular systems, as the unordered structure allows scalable learning over variable-size inputs. For example, EDM (E(n)-equivariant Diffusion Models)[174] generate conformations by learning distributions over point clouds equivariant to Euclidean transformations via diffusion process.

2.1.4.3 Strengths and limitations

Voxel grids offer an intuitive spatial representation but are computationally intensive and sensitive to resolution choices, making them less scalable for large or highly flexible molec-

ular systems. Point clouds, being sparse and equivariant, provide a highly flexible and memory-efficient approach to encoding molecular geometry. They consist of unordered sets of 3D coordinates, with each represented using local features, and avoid discretization altogether. This representation is inherently compatible with permutation invariant and equivariant neural architectures, which are particularly important for modeling physical systems where 3D transformation invariance is essential. Even though point cloud models require specialized neural network designs, they have demonstrated exceptional performance in generative tasks, including the construction of novel 3D molecular conformations, drug-like scaffolds, and even entire protein backbones.

2.2 Deep learning models

2.2.1 Feed forward neural network

Feed forward neural networks (FNNs), sometimes termed as multilayer perceptrons (MLPs), comprise an input layer, one or more hidden layers and an output layer, arranged in a strictly acyclic graph. For an L-layer MLP, the forward pass is

$$\begin{aligned} \mathbf{h}_0 &= \mathbf{x} \\ \mathbf{h}_{l+1} &= \mathrm{ReLU}(\mathbf{W}_l \mathbf{h}_l + \mathbf{b}_l) \\ \hat{y} &= f(\mathbf{h}_L) \end{aligned} \tag{2.1}$$

where $\mathbf{x} \in \mathbb{R}^d$ is a fixed-length molecular feature vector, Rectified Linear Unit (ReLU) is a commonly used non-linear activation function, and f is a task-specific output function (softmax for classification, linear for regression). Model parameters $\mathbf{W}^{(\ell)}, \mathbf{b}^{(\ell)}$ where $\ell = 1 \dots L$ are optimized via stochastic gradient descent (SGD) or its variants on a loss function consistent with maximum likelihood estimation (MLE). For regression tasks, the loss function is typically the mean squared error (MSE). For classification tasks, the cross-entropy loss is employed, reflecting a categorical likelihood.

In early QSAR studies, shallow FNNs coupled with hand-crafted descriptors from software packages such as Dragon[175] and Mordred[176] to predict properties like solubility, logP and bioactivity. A key milestone was the 2012 Merck Kaggle challenge, where multi-task FNNs achieved higher accuracy than random forests using identical input features[177, 178].

In MD analysis, the variational approach for Markov processes network (VAMPNet) employs two weight-sharing FNN that map time-lagged MD frames to soft state assignments[40]. The architecture directly optimizes the VAMP2 score, capturing slow dynamical modes

from high-dimensional MD trajectories. The more recent Memory kErnel Minimization based Neural Networks (MEMnets)[38] move beyond the Markovian assumption in VAMP-Net using IGME[37] theory, by minimizing VAMP2 score together with the time-integrated memory kernel.

To improve generalization and training stability, modern FNN implementations commonly incorporate techniques such as batch normalization and dropout. Batch normalization (BatchNorm) normalizes the input of each layer across a mini-batch, reducing covariate shift which can otherwise slow convergence and destabilize optimization[179]. Dropout[180], a stochastic regularization method, randomly sets a fraction of neuron outputs to zero during training, and effectively prevents co-adaptation and overfitting.

FNNs expect a fixed-length vector and they are inherently non-invariant to input ordering, making them ill-suited for graph-structured or spatial data. Additionally, model performance may be highly dependent on feature selection. Poorly designed input features can result in poor generalization. FNNs are also less interpretable than linear models, though various explainability methods (e.g., SHAP[181], LIME[182]) have been developed to attribute feature importance and enhance model transparency.

Nonetheless, the simplicity, scalability and strong empirical performance of FNNs, especially in low data regimes where feature engineering and domain knowledge plays an important role, make them a foundational component in molecular machine learning. They often serve as the final prediction stage in larger pipelines, processing embeddings generated from preceding convolutional, graph-based, or attention-based layers.

2.2.2 Convolutional neural network

Convolutional neural networks (CNNs) form a deep learning paradigm in which learnable filters are iterated over a structured input tensor to extract local and translationally repeated patterns. First introduced for image recognition, CNNs have been successfully re-purposed for chemical data that can be rendered in one-, two-, or three-dimensional Euclidean arrays.

At the core of a CNN is the convolution operation. Mathematically, given an input tensor $\mathbf{x} \in \mathbb{R}^{C\times H\times W}$ (e.g., a 2D image with height H, width W and C color channels) and a learnable kernel $K \in \mathbb{R}^{C\times k\times k}$, the convolution output at spatial location (i, j) is defined as:

$$(\mathbf{K} * \mathbf{x})_{ij} = \sum_{c=1}^{C} \sum_{u=0}^{k-1} \sum_{v=0}^{k-1} K_{c,u,v} \cdot x_{c,i+u,j+v} \tag{2.2}$$

This operation is repeated over the input using a sliding window mechanism. To preserve spatial resolution, padding is commonly applied at the boundaries. Additionally,

pooling layers (e.g., max or average pooling) are inserted between convolutional layers to reduce spatial dimensionality, promote translational invariance, and control overfitting by downsampling intermediate representations.

A distinctive strength of CNNs is their parameter efficiency. Unlike FNNs where each input unit is connected to every neuron in the next layer, CNNs enforce weight sharing across spatial locations. This drastically reduces the number of learnable parameters and allows the network to generalize learned "local" patterns across space. For example, a fully connected layer for an input flattened from the size of 64×64 would require thousands of parameters, whereas a convolutional layer with a 3×3 kernel requires only 9 parameters per channel if not to include the bias term. This weight sharing allows CNNs to generalize spatial patterns efficiently and makes them highly scalable for large, high-dimensional datasets.

To improve training stability in deeper CNNs, modern architectures such as ResNet[183] often incorporate residual connections. These residual connections alleviate the vanishing gradient problem and enable the construction of very deep networks with improved performance.

CNNs have been successfully adapted to molecular representations of various dimensionalities. 2D molecular images augmented with data transformations have outperformed fingerprint-based models for QSAR prediction tasks[184]. 1D CNNs have been employed to process SMILES strings. This allows the model to capture n-gram–like patterns in SMILES. For example, SMILES convolution fingerprint (SCFP)[185] demonstrated that 1D CNNs can automatically identify functional motifs, such as heteroaromatic rings, thereby eliminating the need for handcrafted descriptors. Similar approaches have been applied to protein and peptide sequences. For example, PeptideModels[186] used 1D CNNs to predict the biological potency of GCGR/GLP-1R dual agonists by modeling local residue interactions directly from amino acid sequences.

3D CNNs extend this framework to volumetric molecular data. AtomNet[169], Atomic Convolutional Neural Networks[187], and KDEEP[170] applied 3D CNNs to voxelized protein–ligand complexes for tasks such as virtual screening and binding affinity estimation.

CNNs provide a powerful and efficient architecture for learning spatial patterns in molecular and biological data. Their applicability spans a range of tasks including property prediction, virtual screening, and structure-function modeling. However, CNNs have several limitations in molecular applications. One major constraint is their reliance on grid-structured input data, which may necessitate costly preprocessing such as voxelization. This is especially inefficient for sparse molecular structures, where the majority of the voxel grid remains empty. Additionally, standard CNNs lack equivariance to spatial rotations and

translations, which can impair their performance in 3D tasks where molecular orientation is arbitrary. These issues have been partially mitigated through data augmentation or addressed directly using symmetry-aware architectures such as SE(3)-Transformers and E(n)-equivariant networks, which explicitly encode physical symmetries into the model.

2.2.3 Recurrent neural network

Recurrent Neural Networks (RNNs) are specifically designed to process sequential data by modeling dependencies across time steps or sequence positions. In contrast to FNNs and CNNs, which process inputs in fixed-size windows, RNNs maintain a hidden state that evolves dynamically as each token in a sequence is processed. This design makes RNNs particularly well-suited for modeling molecules represented as linear sequences such as SMILES strings or peptide sequences, where the order of elements reflects critical structural or functional relationships.

The mathematical formulation of an RNN involves recursively updating a hidden state vector $\mathbf{h}_{t+1} \in \mathbb{R}^d$ based on the input at time step $t+1$, denoted $\mathbf{x}_{t+1}$, and the previous hidden state $\mathbf{h}_t$. This update is typically defined as:

$$\mathbf{h}_{t+1} = f(\mathbf{W}_x \mathbf{x}_{t+1} + \mathbf{W}_h \mathbf{h}_t + \mathbf{b}) \tag{2.3}$$

where f is a nonlinear activation function (commonly tanh or ReLU), and $\mathbf{W}_x$, $\mathbf{W}_h$ and $\mathbf{b}$ are learnable parameters. Final predictions may be generated at each time step or only at the sequence end, depending on the task. To train RNNs, Backpropagation Through Time (BPTT)[188] is employed, which unfolds the recurrent computations and propagates gradients back through each time step. This process allows the network to learn long temporal dependencies, although it is prone to issues such as vanishing and exploding gradients, which have been mitigated in part by Long Short-Term Memory (LSTM)[189] and Gated Recurrent Units (GRUs)[190].

The sequential nature of SMILES and SELFIES strings makes them natural substrates for recurrent modelling, but it also introduces two engineering considerations. First, molecules exhibit variable lengths. Efficient mini-batch training therefore relies on padding each sequence with a special mask token to a fixed maximum length so that the entire batch can be processed as a dense tensor. A companion binary mask ensures that padding symbols neither contribute to hidden state updates nor to the loss function. Second, RNN performance benefits greatly from randomized SMILES enumeration as data augmentation[191]. Because any molecular graph admits many depth-first traversal orders, one can generate multiple syntactically distinct yet chemically equivalent strings per molecule, as discussed

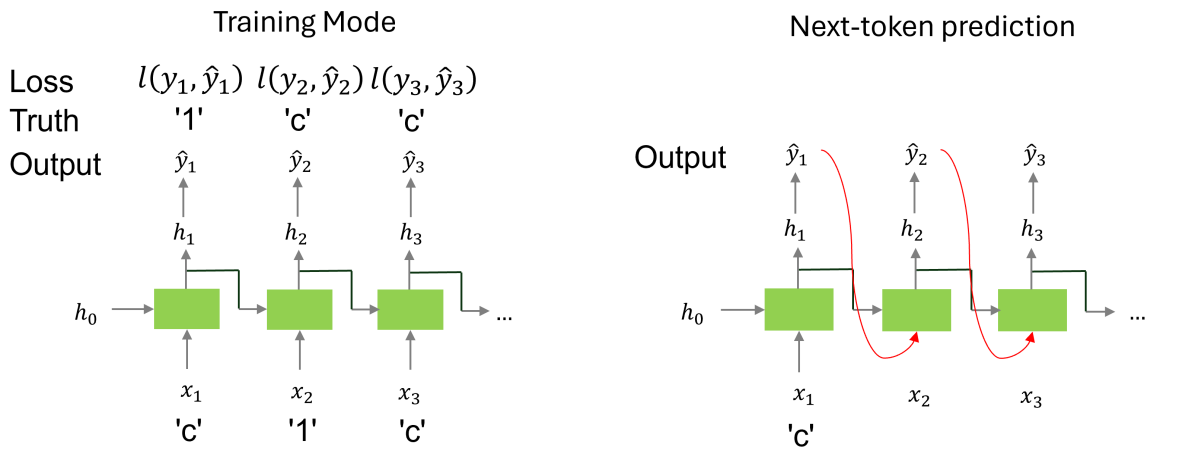


Figure 2.4: Exposure bias arises when the training procedure uses ground-truth tokens while generation relies on self-predicted tokens, leading to a mismatch in input distributions.

in Section 2.1.2.1. Exposure to random SMILES analogously to paraphrasing in natural language processing, improving the model's ability to generalize across syntactic variants.

RNNs have been extensively applied to molecular property prediction and generative modeling. In property prediction tasks, SMILES strings are tokenized and processed by an RNN to model substructure dependencies. For example, SMILES2Vec demonstrated competitive performance with descriptor-based models across standard QSAR benchmarks[133].

RNNs can operate in an autoregressive mode to generate SMILES strings token by token, modeling the conditional distribution $p(x_t|x_{0:t-1})$. During training, the model uses teacher forcing, wherein the ground-truth token x_{t-1} is fed at each time step. At inference, however, the model must rely on its own predictions, which can lead to exposure bias due to distributional mismatch between training and generation phases (Figure 2.4). Scheduled sampling[192] mitigates this issue by gradually replacing true tokens with predicted ones during training to improve robustness.

Control tokens such as SOS (start-of-sequence), EOS (end-of-sequence), and PAD (padding) are critical in sequence modeling. SOS and EOS delimit sequences during generation, allowing models to produce variable-length molecules, while PAD tokens enable batch-level alignment. Proper handling of these tokens is essential for effective supervised learning and controlled generation.

RNN-based generative models have been applied to reaction prediction, where the

input sequence represents reactants and reagents, and the output sequence represents the predicted product. Notably, sequence-to-sequence (Seq2Seq) RNNs with attention mechanisms were instroduced to translate reactant SMILES into product SMILES, modeling chemical reactions as a language translation problem[193, 146].

RNNs have several limitations despite of their broad applicability. Their sequential nature makes them inherently slow to train and difficult to parallelize, especially for long sequences. They are also vulnerable to gradient instability and may struggle to capture long-range dependencies without architectural modifications. While LSTM and GRU variants partially address these issues, newer architectures such as Transformers have begun to replace RNNs in many applications due to their superior scalability and ability to model global relationships more efficiently. Moreover, linearization of molecular graphs into strings can obscure global topological features, motivating hybrid models that couple a recurrent decoder to graph-based encoders. Nonetheless, RNNs retain important advantages in low-data regimes. Their compact parameterization, natural handling of variable-length sequences, and conceptual simplicity make them well-suited for interpretable baselines and modular components within larger molecular deep learning pipelines.

As you might have noticed, both RNNs and 1D CNNs are widely used for sequence-based modeling in cheminformatics and bioinformatics, particularly for predictive tasks involving SMILES strings and peptide chains. While these architectures are designed for ordered inputs, their computational and inductive properties differ markedly. 1D CNNs extract localized patterns such as functional groups or conserved motifs by convolving filters along the input sequence. In contrast, RNNs model sequential dependencies by maintaining a recurrent hidden state, enabling them to capture broader contextual relationships and long-range interactions across the entire sequence. CNNs are computationally efficient due to weight sharing and enable high-throughput parallelism, while their ability to model long-range dependencies is limited by the receptive field, which depends on kernel size and network depth. In contrast, RNNs are inherently sequential in both training and inference, often making them slower and more difficult to scale than CNNs. Empirical results have shown that combining CNNs with RNNs achieved stronger predictive power than architectures only containing RNNs, as illustrated in the SMILES2Vec paper[133] paper. More importantly, CNNs are not typically used for molecular generation. Their lack of an inherent temporal framework makes them ill-suited for generating sequences token by token. This role is more naturally filled by RNNs or Transformers, which can model conditional dependencies and control sequence flow via autoregressive sampling.

2.2.4 Graph neural network

Graph Neural Networks (GNNs) extend deep-learning principles to non-Euclidean domains by operating directly on graph-structured data. In a molecular context, nodes represent atoms (or residues) and edges encode chemical bonds, together with other information such as spatial proximity and bond angles. This makes GNNs ideally suited to capturing topological and structural features that are difficult to encode in grid-based or sequential formats.

The foundational framework for GNNs is the message-passing neural network (MPNN) paradigm[194]. At each layer, a node v collects information from its neighbours $\mathcal{N}(v)$, aggregates the incoming messages, and updates its hidden state $\mathbf{h}_v^{i+1}$:

$$\begin{aligned} \mathbf{m}_v^{i+1} &= \sum_{u \in \mathcal{N}(v)} \phi_m(\mathbf{h}_v^i, \mathbf{h}_u^i, \mathbf{e}_{uv}) \\ \mathbf{h}_v^{i+1} &= \phi_u(\mathbf{h}_v^i, \mathbf{m}_v^{i+1}) \end{aligned} \tag{2.4}$$

Here ϕ_m and ϕ_u are neural networks for message propagation and updating and $\mathbf{e}_{uv}$ denotes an incoming edge feature vector from u to the central node v. Several GNN variants have been introduced to improve performance and model expressivity. For example, Graph Attention Networks (GATs)[195] incorporate attention-weighted summation rather than the uniform summation, to adaptively prioritize neighbor contributions based on learned importance scores. SchNet[196] leverages continuous-filter convolutions to integrate spatial distances into the message passing, and achieves $E(3)$-invariance with respect to translation and rotation.

In practical settings, GNN frameworks such as PyTorch Geometric batch graphs of variable sizes by concatenating their adjacency structures into a single sparse matrix and offsetting node indices with batch identifiers. This strategy affords efficient GPU utilization on datasets comprising thousands of molecules.

GNNs offer significant advantages over sequence-based or grid-based models. Unlike 1D CNNs or RNNs, which impose linear orderings on input data, GNNs are intrinsically permutation-invariant and respect the native topology of molecular graphs. GNNs provide a unifying framework for learning from structured molecular and biological data. Their ability to model topological relationships and respect graph symmetries position them as foundational tools in modern cheminformatics, structural bioinformatics, and the modeling of dynamical molecular systems.

Despite their strengths and versatility, GNNs face certain limitations. Deep GNNs may suffer from oversmoothing[197], where node representations become indistinguishable after many layers, degrading model expressivity. Computational scalability is another

concern, particularly when modeling large biomolecules[198] or dense interaction graphs. Furthermore, even though GNNs are permutation-invariant, they are not inherently equivariant to 3D transformations. This limits their application in tasks such as predicting force field and molecular docking. Continued innovations in equivariant architectures[199] promise to expand their applicability. For example, E(n)-equivariant GNNs[200] and SE(3)-Transformers[201] have been proposed which preserve equivariance under Euclidean group operations.

2.2.5 Transformers and large language models

The Transformer architecture[202], originally proposed for natural language processing (NLP) tasks, has emerged as a foundational model in deep learning across diverse domains. Unlike traditional RNNs, Transformers eliminate recurrence and instead rely on a self-attention mechanism, which models dependencies between all tokens in a sequence in parallel. Moreover, unlike convolutional neural networks (CNNs) that rely on local spatial correlations, Transformers compute global pairwise correlations, allowing them to model long-range dependencies more effectively.

A critical distinction between Transformers and earlier sequence models lies in their treatment of order. RNNs inherently preserve sequence position through their recurrent architecture, whereas Transformers, due to their fully parallel attention mechanism, lack a built-in notion of token order. To address this, Transformers employ positional encoding to inject information about the relative or absolute position of each token into its embedding vector. In the original Transformer model[202], sinusoidal positional encodings were used, defined as a deterministic function of position and embedding dimension:

$$\begin{aligned} PE_{(pos,2i)} &= \sin\left(\frac{pos}{10000^{2i/d}}\right) \\ PE_{(pos,2i+1)} &= \cos\left(\frac{pos}{10000^{2i/d}}\right) \end{aligned} \tag{2.5}$$

where pos denotes the position and i denotes the dimension index. In later variants, learned positional embeddings replaced sinusoidal encodings, allowing the model to optimize positional representations jointly with other parameters. More recently, rotary positional encoding (RoPE)[203] has been introduced to improve the modeling of relative positional information in addition to the global positional information. Positional encodings ensure that sequence orders are preserved in the learned representation.

A key innovation of the Transformer is multi-head attention (MHA), which enables the model to jointly attend to information from different representation subspaces. In a MHA block of h heads (Figure 2.5), the k-th attention head takes $h_i^k \in \mathbb{R}^{l \times d_k}$ as the input,

where l is the length of the sentence and d_k is the hidden dimension of the attention head, and $d_k = d/h$. The h_i^k is projected to the query (Q^j), key (K^k), and value (V^k) matrices $\in \mathbb{R}^{l \times d_k}$ via linear transformation with learnable parameters $W_Q^k, W_K^k, W_V^k \in \mathbb{R}^{d_k \times d_k}$:

$$Q^k, K^k, V^k = h_i^k W_Q^k, h_i^k W_K^k, h_i^k W_V^k \tag{2.6}$$

After that, the attention scores can be computed via scaled dot-product attention and softmax normalization:

$$S^k = Softmax(\frac{Q^k (K^k)^T}{\sqrt{d_k}}) \tag{2.7}$$

S^k is a row-normalized matrix of shape $l \times l$ where $\sum_{j=0}^{l-1} S_{ij}^k = 1$. Each entry S_{ij}^k in the score matrix denotes the contribution factor from the token j to token i. The output of the k-th attention head $h_{i+1}^{k\prime}$ is a weighted sum of the value vectors, producing a new representation for each token after attending to the entire sequence:

$$h_{i+1}^{k\prime} = S^k V^k \tag{2.8}$$

The output of MHA block $h_{i+1} \in \mathbb{R}^{l \times d}$ is a concatenated form of all attention head outputs along the d_k dimension, with a residual connection:

$$\begin{aligned} h_{i+1}^{\prime} &= Concat_k(h_{i+1}^{k\prime}) \\ h_{i+1} &= f(h_{i+1}^{\prime}) + h_i \end{aligned} \tag{2.9}$$

The feed forward neural network projects the output of MHA $h_{i+1} \in \mathbb{R}^{l \times d}$ to an intermediate state of higher dimension (usually $\in \mathbb{R}^{l \times 4d}$), applies a non-linear activation, and finally reduces the intermediate state back to $\mathbb{R}^{l \times d}$.

Transformers are computationally demanding. The standard self-attention mechanism scales quadratically with sequence length. This becomes a significant bottleneck in modeling long sequences, such as high-resolution protein structures, long SMILES strings, or multi-chain protein complexes.

To address these challenges, several architectural refinements have been proposed. Sparse attention mechanisms compute only a subset of attention scores, reducing computational complexity from $O(n^2)$ to $O(n \log n)$ or even linear in some cases. Models such as Longformer[204] and BigBird[205] implement sparse attention mechanism that preserve the ability to capture long-range dependencies while remaining scalable. Another emerging class of models uses latent attention to reduce input dimensionality. For instance, Perceiver[206] introduces a fixed-size latent array that interacts with the input via cross-

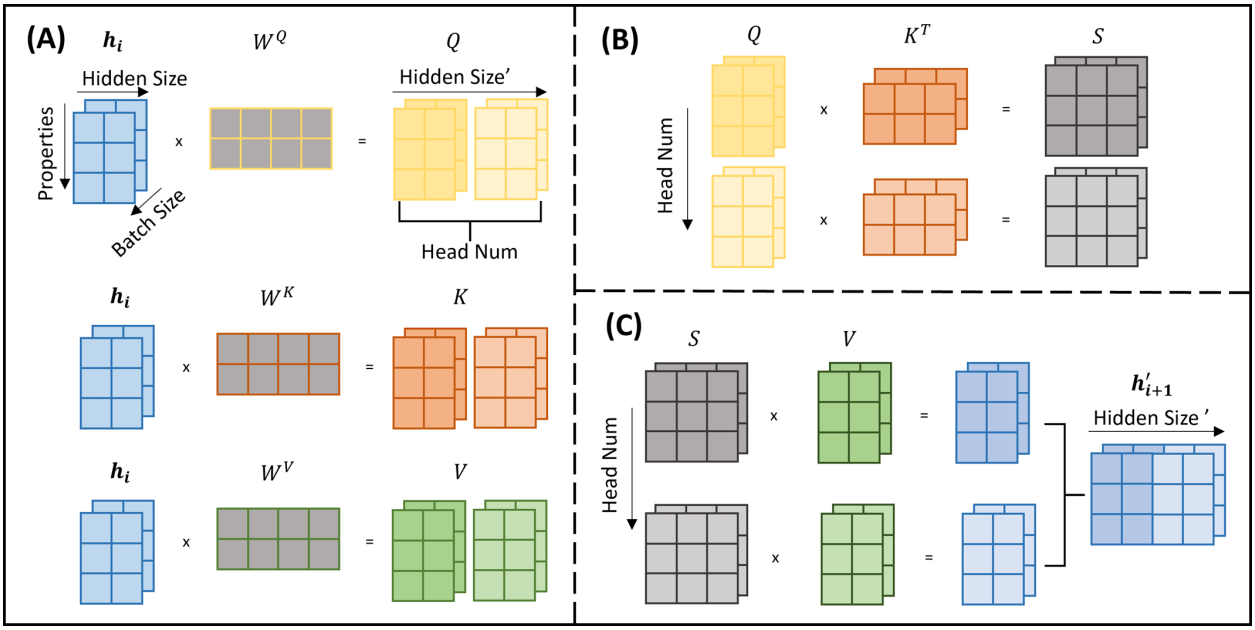


Figure 2.5: Multi-head attention mechanism of Transformer. (A) Project input h_i to Q,K,V matrices. Input h_i has the shape of (l, d), where l represents the length of sentence and d is the dimension of each word. Specifically, $l = 81$ in FeatureDock[2] containing 80 physicochemical features and the class token. Q,K,V has the shape of (l, d) each, where d is the dimension of hidden state. (B) Calculate multi-head attention weight matrices $S \in \mathbb{R}^{l \times l}$. Each head processes a part of the Q,K,V matrix. (C) Concatenate results from multiple heads and output h'_{i+1}. h'_{i+1} will then be fed into a fully-connected layer and combined with h_i via residual connection to generate h_{i+1}.

attention, enabling processing of arbitrarily large inputs without quadratic cost. These latent architectures are particularly promising for handling large chemical graphs or dense voxel grids in molecular modeling.

Transformer-based models have been adopted in domains beyond NLP, including computer vision, where architectures such as the Vision Transformer (ViT)[207] and Swin Transformer[208] have outperformed classical CNNs on benchmark image classification tasks. Similarly, the flexibility of attention-based architectures has catalyzed their use in drug discovery and biomedical modeling. Transformers have been used to model drug–target interactions (DTIs)[209] by jointly learning from molecular and biological sequence representations.

Protein language models (PLMs) such as ProtBert[210] and ESM-2[17] learn rich embeddings of amino acid sequences, supporting downstream tasks. In FeatureDock[2], a Transformer encoder was employed to learn spatially dependent interactions across physicochemically embedded grid points in the protein pocket. These results underscore

the flexibility of attention mechanisms in modeling multiple levels of biological complexity.

Transformers are also central to the rise of large language models (LLMs) in scientific research. For example, by leveraging task-specific prompts and pretraining on curated biological datasets, BioT5[211, 212] demonstrates strong transfer learning across a wide spectrum of protein-related problems, including protein language modeling, secondary structure prediction, and contact prediction tasks.

Beyond static prediction, LLMs are increasingly deployed as autonomous agents in scientific workflows, facilitating end-to-end research workflows[213]. Especially, LLMs are rapidly advancing toward serving as coding agents in scientific research. While general-purpose coding agents like Codex[214] and Code LLaMA[215] have demonstrated impressive performance in software engineering tasks, their application to scientific domains remains limited without further adaptation. In contrast, scientific coding agents are either fine-tuned on domain-specific code bases or guided via structured prompts that embed scientific context. These agents are designed to understand the terminology, logic and toolchains native to the research domains. This distinction is empirically confirmed in BioCoder[216], BioAgent[217] and ScienceAgentBench[3] by showing that scientific agents outperform general agents when evaluated on real-world scientific problems.

ScienceAgentBench has also demonstrated that prompt engineering is more critical than domain knowledge alone in determining the performance of scientific agents. Figure 2.6 provides a comparison of success rates (SR) across various large language models (LLMs) using three prompting frameworks (Direct Prompting, OpenHands CodeAct, and Self-Debug) and two domain knowledge settings (with or without expert-provided knowledge), evaluated on the 102 scientific tasks in the benchmark dataset. Remarkably, even the simplest self-debug prompting strategy yielded greater performance gains than the OpenHands coding agent with domain knowledge.

Prompt engineering is critical not only for encoding scientific assumptions and goals but for constraining the agent's reasoning within domain-relevant bounds. Agent engineering techniques such as modular planning, sub-agent delegation, tool integration, multi-agent collaboration, and self-refinement protocols have shown to dramatically improve the reliability and reproducibility of scientific outputs. Without these engineered scaffolds, even powerful LLMs tend to generate brittle or incomplete workflows when faced with complex scientific queries.

Ultimately, the goal is to engineer scientific agents that participate in a closed-loop system[218], including reading literature, identifying gaps, formulating questions, designing *in silico* or *in vitro* experiments, analyzing results, and feeding insights back into the loop. This requires agents not only to access external tools and data but to maintain persistent

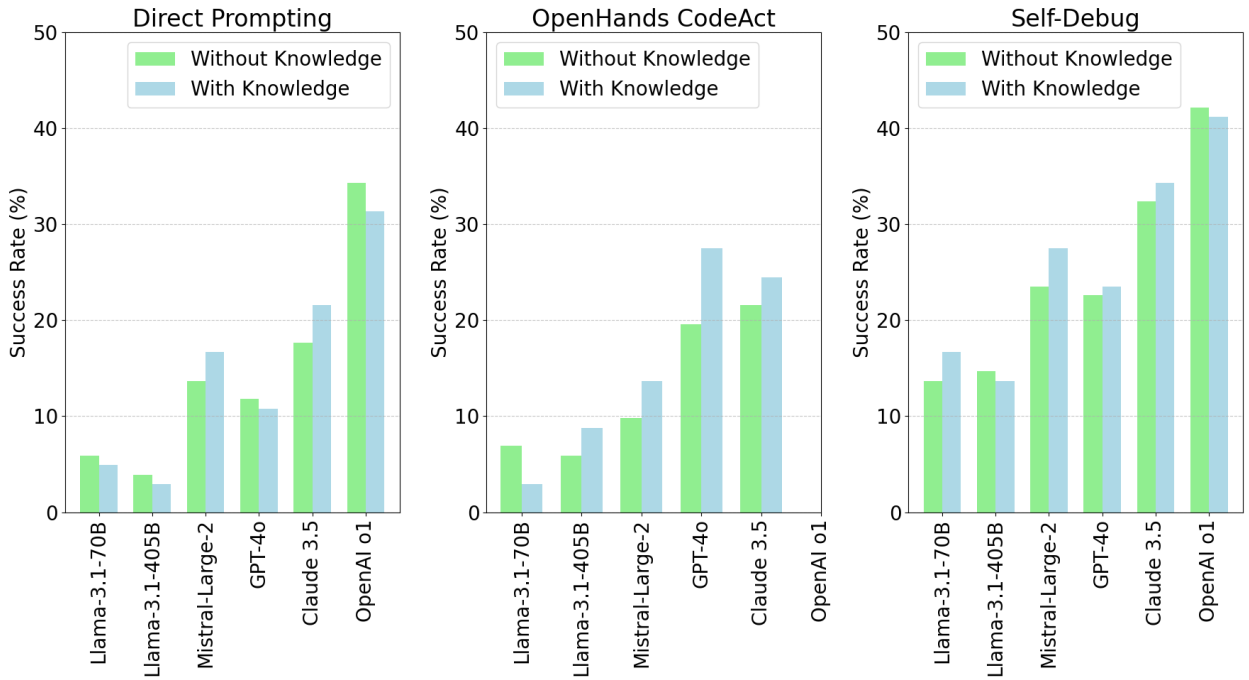


Figure 2.6: Comparison of success rates (SR) across frameworks for different language models on 104 high-quality scientific coding tasks, with values adapted from ScienceAgentBench[3] Table 3. Each subplot shows performance under Direct Prompting, OpenHands CodeAct, and Self-Debug frameworks, with and without expert-provided domain knowledge. Claude 3.5 and OpenAI o1 consistently outperform other models, particularly under the Self-Debug framework. OpenHands results are unavailable for OpenAI o1.

memory, adapt goals based on feedback, and exhibit meta-cognitive behaviors such as uncertainty estimation and hypothesis revision. By unifying reasoning, prompt engineering, and multi-agent collaboration, we move closer to a system capable of autonomously generating publishable scientific knowledge.

The rapid evolution of Transformer-based architectures has fundamentally reshaped the computational chemistry and biology. Transformers, LLMs and LLM agents are constituting a new research paradigm where deep learning not only analyzes and generates data but also actively participates in the design, execution, and interpretation of scientific discovery.

2.2.6 Generative models

Generative models aim to learn the the data distribution $p(\mathbf{x})$ or the joint distribution $p(\mathbf{x}, y)$, enabling the synthesis of novel and meaningful samples. In molecular science, these models have proven transformative for tasks such as *de novo* molecular generation,

property-conditioned molecular generation, conformational sampling, and kinetic pathway discovery.

Among the different generative model architectures, the Variational Autoencoder (VAE)[219, 220] stands out for its ability to learn a continuous and controlled latent space that supports sampling and generating. A VAE consists of two neural networks. An encoder maps an input $\mathbf{x}$ to a distribution over latent variables $q_\phi(\mathbf{z}|\mathbf{x})$, and a decoder reconstructs the input from a latent vector $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$, sampled from this distribution. The training objective is to maximize the evidence lower bound (ELBO) on the marginal likelihood $\log p_\theta(\mathbf{x})$ given by:

$$\mathcal{L}(\theta, \phi; \mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - \mathrm{KL}\left(q_\phi(\mathbf{z}|\mathbf{x}) \,\|\, p(\mathbf{z})\right) \tag{2.10}$$

A key technical challenge in training VAE is that the encoder outputs a distribution over the latent variables $\mathbf{z}$, rather than a single deterministic vector. As a result, the direct sampling of $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$ within the computational graph would introduce a non-differentiable operation, making it impossible to propagate gradients through the sampling process during training. To resolve this, VAEs employ the reparameterization trick, which transforms the stochastic sampling operation into a deterministic, differentiable function. Specifically, if the encoder outputs a Gaussian distribution with mean μ and diagonal covariance σ^2, the latent variable is sampled via:

$$\begin{aligned} \mathbf{z} &= \mu + \sigma \odot \epsilon \\ \epsilon &\sim \mathcal{N}(0, \mathbf{I}) \end{aligned} \tag{2.11}$$

Here, $\odot$ denotes element-wise multiplication, and ϵ is a noise vector drawn from a standard normal distribution. This reparameterization isolates the stochastic component from the trainable parameters, allowing gradients to flow through μ and σ during back-propagation.

Compared to Autoencoders (AEs)[221] which encode inputs into deterministic vectors and optimize purely for reconstruction accuracy, VAEs introduce stochasticity and a KL divergence regularization term, which encourages the latent space to follow a smooth, continuous distribution. This property makes VAEs particularly well-suited for interpolation, sampling, and property-conditioned generation. AEs are often better suited for denoising or compression tasks, while VAEs provide stronger support for generative tasks. For example, VAEs have been widely used for SMILES string-based molecular generation and offers a latent optimization framework where molecular properties can be tuned by navigating the latent space[135].

Beyond small molecule design, VAEs have employed to model kinetics processes and conformational transitions. For example, by training the VAE on kinetically weighted pathway images and clustering the latent embeddings, the latent-space path clustering (LPC)[108] algorithm identified four metastable channels in hydrophobic aggregation and two dominant routes in WW-domain folding.

The landscape of deep generative modeling has expanded far beyond the pioneering variational autoencoder (VAE). While VAEs remain valuable for their mathematically principled latent spaces and tractable likelihoods, recent advances have introduced more expressive and flexible alternatives. Emerging classes of generative models such as diffusion models[222, 223] progress beyond VAEs, retaining a continuous latent trajectory while overcoming the Gaussian prior assumption in VAE latent space. Flow-matching frameworks[224] generalize over the continuous normalizing flows[225] by training velocity fields that match exact likelihoods, closing the gap between statistical rigor and generative quality. Collectively, these innovations are reshaping drug-discovery pipelines[226, 153].

3 Discover Inhibitive Fragments using Deep Learning Methods

As discussed in chapter 1 that fragment-based drug discovery (FBDD) helps efficiently reduce the chemical space, this chapter focuses on the ChemPLAN-Net framework (Figure 3.1), a deep learning framework developed to predict binding fragments for protein binding sites. It is partially adapted from "*ChemPLAN-Net: A deep learning framework to find novel inhibitor fragments for proteins*". Suarez Vasquez, M. A., Xue, M., Lam, J. H., Goonetilleke, E. C., Gao, X., Huang, X. (2021).

The ChemPLAN-Net framework was originally introduced by Dr. Michael Suarez. He applied this framework to the protease system including HIV-1 protease and SARS-CoV-2 main protease (M^{pro}). The sections including 3.2.2, 3.2.3 and 3.3.1 are adapted and rewritten based on his results published in the ChemPLAN-Net manuscript.

The present chapter extends the original ChemPLAN-Net work in several significant directions. My contribution include developing and refining the dataset curation pipeline, training ChemPLAN-Net to study kinase inhibitive fragments and molecular glue systems, using predicted fragments to find candidates from a virtual synthesized compound library, applying the deep learning based fragment linking method DeLinker to the ChemPLAN-Net predictions for whole ligand design. The sections including 3.2.1, 3.3.2, 3.3.3, 3.3.4, 3.4, 3.5 are my follow-up work based on ChemPLAN-Net, but not included in the published results.

3.1 Previous work

This section reviews two methods, FragFEATURE[227] and EnsFragFEATURE[228], which formed the foundation for ChemPLAN-Net. These approaches exemplify early attempts to link physicochemical environments with fragment-level binding preferences using traditional machine learning methods. Their methodologies, assumptions, performance characteristics, and limitations are discussed to contextualize the need for deep learning-based extensions.

3.1.1 FragFEATURE: Knowledge-Based Fragment Binding Prediction

FragFEATURE developed a data-driven approach to predict potential fragments of protein binding sites using local environments and the k-nearest neighbour algorithm. The

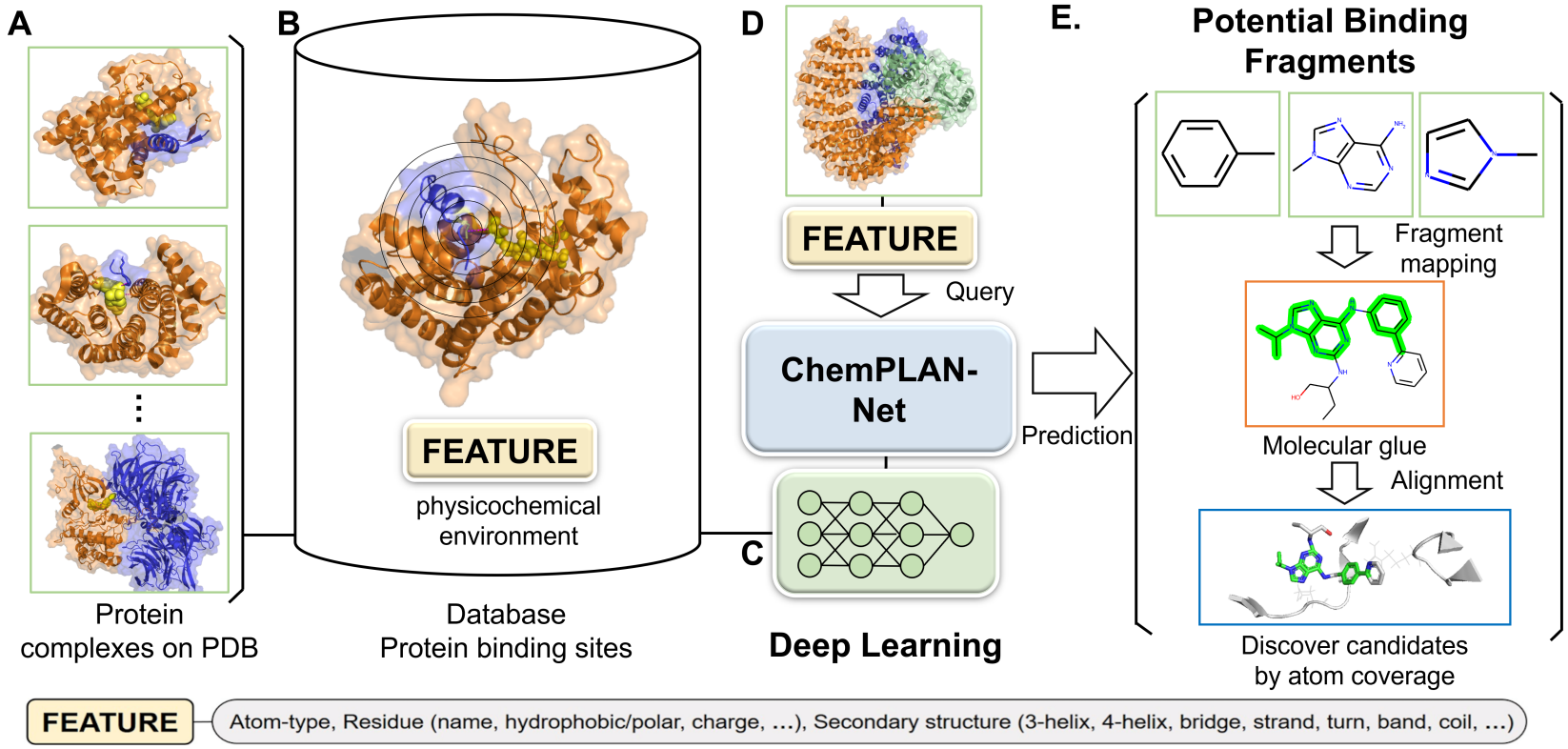


Figure 3.1: ChemPLAN-Net as an engine for fragment-based drug discovery. (A) Compile a structure dataset from the RCSB PDB databank containing ternary complexes of a target of interest. (B) Extract the FEATURE vectors from potential binding sites, and construct a training dataset containing binding and non-binding FEATURE vector-fragment pairs based on a predefined fragment library. (C) Train ChemPLAN-Net on the dataset. (D) Once the deep neural network has been trained, queries can be made to identify matching ligand fragments from our compound library using protein *apo* structures. (E) The output ligand fragments will be mapped back to compounds in library and select candidate compounds based on atom coverage ratios.

assumption of FragFEATURE is that proteins may bind to similar fragments due to sharing similar local environments, even though the whole sequence or global structure might differ.

The method constructs a knowledge base by mining protein–ligand cocrystal complexes from the Protein Data Bank (PDB)[229]. Each local binding environment is defined around a functional center and encoded using the FEATURE vector descriptors. At the time of publication, FragFEATURE had assembled approximately 1.7 million FEATURE vectors across 23 types of functional centers, extracted from 34,000 protein–ligand complexes. The details about functional centers will be discussed more in Section 3.2.1.3. Ligands were decomposed into overlapping substructures by screening against a curated library comprising 225,000 fragments, each containing three to thirteen heavy atoms collected from PubChem. In total, 250 million local environment-fragment binding pairs were recorded, capturing both common and rare interaction motifs.

To predict fragment binding for a new protein structure, FragFEATURE encodes candidate binding sites as FEATURE vectors and compares them to non-homologous vectors in the knowledge base. The similarity between environments is quantified using a background-weighted Tanimoto coefficient, and the five nearest neighbors are retrieved via a k-nearest neighbors (k-NN) search. Fragment enrichment is then evaluated using statistical metrics, including hypergeometric and Fisher's exact test p-values, to rank predicted fragments by significance.

In benchmark evaluations, FragFEATURE achieved a fragment recovery precision of 74% and recall of 82%, demonstrating high performance in retrieving native ligand substructures. A key strength of this method lies in its reliance on local descriptors rather than global sequence or structural similarity, enabling robust performance even for rare or structurally divergent binding sites.

FragFEATURE was the first framework to couple high-dimensional environmental descriptors with large-scale statistical learning across the entire PDB. By formalizing fragment preference as a k-NN retrieval problem and validating its transferability across thousands of pockets, this study laid the empirical foundation upon which subsequent machine- and deep-learning approaches, such as EnsFragFEATURE and ChemPLAN-Net.

3.1.2 EnsFragFEATURE: Fragment Binding Prediction from Conformational Ensembles

EnsFragFEATURE extends the original FragFEATURE algorithm by incorporating large-scale MD sampling and h a dramatically faster instance-retrieval algorithm.

In EnsFragFEATURE, an ensemble of *apo* Cyclin Dependent Kinase 2 (CDK2) conformers was first generated by MD simulation, seeded from Collective Molecular Dynamics (CoMD) results. Each potential ligand-contacting functional center in the ensemble was encoded using FEATURE vector, the same as in FragFEATURE. By applying the more efficient ball trees[230] to the knowledge base curated in FragFEATURE, the complexity of retrieval is reduced from $O(N^2)$ to $O(NlogN)$, yielding a thirteen-fold acceleration on average for fragment retrieval queries.

For every retrieved candidate fragment, hypotheses gathered across the ensemble were collated into site-wise histograms and filtered with a Benjamini–Hochberg hypothesis that controlled the false-discovery rate (FDR), thereby converting raw hit counts into statistically significant propensities. EnsFragFEATURE further utilized Peacock's two-sample Kolmogorov-Smirnov test to analyze whether the binding fragments are consistent across ensemble conformers.

Results indicated that certain fragments exhibited stable binding preferences across conformers, while others varied significantly depending on the sampled protein conformation. This highlights the importance of conformational change in fragment-based binding prediction. By coupling large-scale conformational sampling with efficient instance retrieval and robust statistical inference, EnsFragFEATURE offers a more comprehensive and biologically realistic model of fragment binding behavior.

3.2 ChemPLAN-Net: A deep learning framework to find novel inhibitor fragments for proteins

ChemPLAN-Net replaces the retrieval-based paradigm of FragFEATURE and EnsFragFEATURE with a supervised deep learning model that learns the compatibility between protein environments and ligand fragments. It accepts a paired query as the input, containing the FEATURE vector that describes the local environment of a binding site and the 1,024-bit Morgan fingerprint that encodes a candidate fragment. By learning a joint latent space that embeds both protein and chemical representations, ChemPLAN-Net enables to explore the fragment space, beyond the scope of FragFEATURE and EnsFragFEATURE. Furthermore, the data generation process underlying ChemPLAN-Net has been restructured and optimized relative to FragFEATURE, yielding a more efficient, extensible, and chemically expressive pipeline.

3.2.1 Dataset curation

The dataset curation pipeline inherits the foundational principle from FragFEATURE[227] by combining protein microenvironments and ligand-derived fragments , but the ChemPLAN-Net-Data-Generation pipeline (Figure 3.2) has been substantially generalized in three ways:

- **Flexible fragment library curation**. FragFEATURE curated a static fragment library by collecting low molecular weight (MW) compounds from PubChem[90], followed by computationally intensive duplicate removal. Afterwards, FragFEATURE decompose each ligand by screening against this predefined fragment library and recording substructure hits. In contrast, ChemPLAN-Net preserves the substructure screening route while introducing an additional SMARTS-based bond-cleavage mode. This allows users to define chemical rules (e.g., "non-aromatic single bonds not in rings") to generate context-specific fragment libraries, thereby enabling the incorporation of domain knowledge or target-specific chemical intuition. Detailed strategies for fragment library construction are further discussed in Section 3.2.1.4.

- **Rich fragment representations** Each fragment is recorded as a canonical SMILES string rather than PubChem ID used previously, which can be seamlessly converted into multiple machine-learning ready formats discussed in Chapter 2.1, including extended connectivity fingerprints (ECFP), molecular graphs, or other bag of properties. After this change, this dataset curation pipeline enables to not only provide protein features and fragment fingerprints as ChemPLAN-Net inputs, but also make ChemPLAN-Net extensible for other fragment representations, thus enabling models to reason over chemical similarity rather than treating every fragment as independent.

- **User-friendly Python-ecosystem implementations**. The entire pipeline is implemented in Python 3 and leverages actively maintained packages such as RDKit[231] , BioPython[232], Open-Source PyMOL, and PubChemPy for cheminformatics and bioinformatics tasks. This removes the legacy Python 2.7 scripts and external Java executables (e.g., SMSD toolkit[233]) used by FragFEATURE. The streamlined software environment improves reproducibility, maintainability, and accessibility for the research community.

Collectively, these enhancements make this dataset curation engine both chemically expressive and practically scalable. By enabling the systematic pairing of diverse protein microenvironments with chemically rich fragment libraries, the pipeline ensures high-quality training data for deep learning models designed to infer fragment binding propensities.

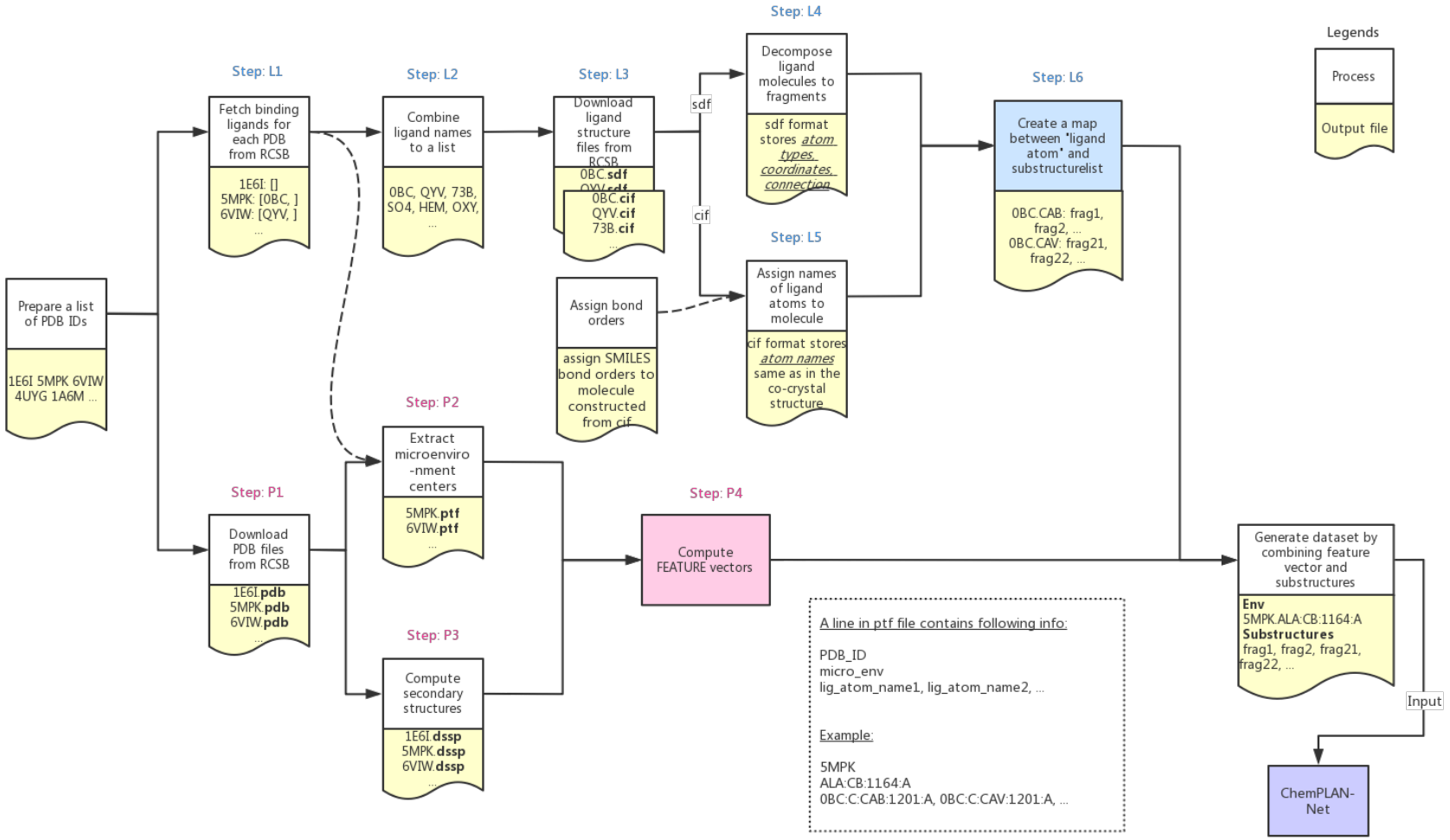


Figure 3.2: Workflow of dataset curation for ChemPLAN-Net. The steps for processing ligands as fragments are labeled in blue with prefix starting by the letter "L" (Ligand). The steps for extracting local environments from protein functional centers are labeled in red with prefix starting by the letter "P" (Protein).

3.2.1.1 Prepare PDB IDs for cocrystal structures of interest

The data generation pipeline starts by preparing a list of PDB IDs corresponding to protein–ligand cocrystal structures relevant to the biological target class. Such a list can be efficiently prepared by searching Enzyme Commission number (EC number) or sequence similarity in RCSB PDB When preparing for proteases and kinases. After a list of protein-ligand complexes are prepared to query from the RCSB Protein Data Bank (PDB), only structures solved by X-ray crystallography with high resolution (e.g.$\leqslant$ 3.0Å) are kept for downstream process.

However, for more complex binding scenarios such as ligands that interact at PPIs, additional filtering criteria must be applied. The following criteria were used to extract a high-quality set of PDB entries containing ligands bound at PPIs, based on a snapshot of the RCSB PDB as of May, 10, 2022:

- Protein structures resolved at a resolution of $\leqslant$ 4.0 Å.
- Structures with a minimum chain length of at least 30 amino acids.
- Presence of at least one non-trivial ligand.
- The bound ligand must be in proximity to at least two non-identical protein chains.

This filtering pipeline resulted in a PPI-ligand dataset containing ~2000 structures. This filtered dataset can couple with the iPPI-DB[234] dataset to further divide into PPI inhibitors and PPI enhancers/stablizers when considering molecule mechanism of action (MMoA).

To assess the completeness of this dataset in capturing known molecular glue structures, we conducted a recovery analysis (Table 3.1) against the 18 reported molecular glue cocrystal structures[1], among which 15 were recovered by this filtering process. The three missed structures were either filtered out due to containing shorter chains or lower resolution. This indicates the high quality of the PPI-ligand dataset collected by this pipeline.

3.2.1.2 Data fetching and preprocessing

After downloaded protein-ligand complex structures of given PDB IDs, The following data preprocessing pipeline further resolves symmetry artifacts, eliminates trivial ligands and coordinate redundancies, and defines consistent functional centers for physicochemical feature extraction.

Table 3.1: Result of the recovery check using 18 PDB structures containing molecular glues fetched from the paper[1].

PDB ID	Recovered by pipeline
6NTS	True
3KMZ	True
2P1Q	False
6UML	True
5HXB	True
6XK9	True
6H0F	True
6H0G	False
7BQU	False
7BQV	True
6PAI	True
6Q0R	True
6UD7	True
6Q0W	True
6TD3	True
6M91	True
5T35	True
6BN7	True

- **Dealing with symmetry and downloading biological assembly**. The primary coordinates from the PDB correspond to the crystallographic asymmetric unit, which requires additional processing to reconstruct the biological assembly based on metadata information in pdb files. If the asymmetric unit already contained multiple equivalent copies of a ligand-bound protein, it is supposed to be treated as a single unique environment to avoid redundant analysis. Crystallographic symmetry is handled carefully by downloading the "Biological Assembly" coordinate files rather than the raw "Asymmetric Unit" [229]. This ensures that biologically relevant quaternary structures are retained for fragment binding context.

- **Filtering and removal of trivial ligands**. After downloading, PDB files are cleaned up by removing trivial ligands, including water (HOH), buffer components, crystallographic additives, common post-translational modifications (e.g. N-acetyl-D-glucosamine, NAG). By excluding such entities, the remaining structures focus exclusively on drug-like ligand interactions rather than crystallization artifacts. This step is consistent with the original FragFEATURE protocol but is reimplemented entirely in the modern Python 3 ecosystem using reproducible modules like RDKit and Biopython, replacing legacy Python 2 scripts.

- **Removal of coordinate duplication**. Each PDB file is processed to standardize coordinates and remove redundancy. If a PDB entry contains multiple models, only the first model was retained for consistency. Atomic positions corresponding to alternate locations are dealt by retaining only the most occupied conformation for each residue. For example, if a residue had alternate conformations (ALTLOC records) labeled "A", "B", etc., the conformation with the highest occupancy was kept as the representative and all others were discarded. Hydrogen atoms were omitted, and only heavy atoms with elements C, N, O, S, or P were retained from the protein structure. More advanced coordinate cleaning procedures can be incorporated in future iterations, such as restoring missing heavy atoms or residues and sampling plausible conformations for incomplete structural regions.

3.2.1.3 Protein local environment extraction

To systematically describe the chemical environment in proteins, we followed the 23 functional centers defined in FragFEATURE (Table 3.2), which capture key interaction sites. Among these, 21 centers are derived from side chains and 2 from backbone atoms. Most centers correspond to individual atoms (e.g., LYS.NZ, CYS.SG), while others represent pseudocenters, defined as geometric centroids of functionally coherent atom sets (e.g., PHE.PSEU denoting the center of the phenyl ring). Two canonical labels (RES.N and RES.O) unify backbone amide nitrogens and carbonyl oxygens respectively, across all residues. These centers serve as anchors for extracting spatially localized physicochemical features relevant to fragment binding.

Each functional center was extracted from the cleaned protein structure and retained if located within its defined cutoff distance from any ligand heavy atom. For each such center, FEATURE vector descriptors[235] listed in Table 3.3 are used to quantitatively characterize physicochemical properties of different hierarchies (e.g., negative and positive charges, partial charges, atom types, residue types, the secondary structure, hydrophobicity, etc.) from its local environment.

Each local environment is defined as a sphere with a radius of 7.5Å centered on the functional center. This sphere is subdivided into six concentric shells, each 1.25Å thick. Within each shell, a fixed set of 80 physicochemical properties is computed, yielding a final descriptor of 480 (6×80) dimensions. The number of shells and shell widths are customize parameters of the FEATURE program. We used the same parameters as in FragFEATURE and NucleicNet[171] as they have been proven sufficient to describe protein local environments for identifying both protein-ligand and protein-RNA interactions.

Table 3.2: Functional centers, atom components in each functional center and distance cutoffs

Name	**Atom components**	**Cutoff (Å)**
ALA.CB	ALA.CB	5.0
ARG.CZ	ARG.CZ	5.0
CYS.SG	CYS.SG	5.0
ILE.CB	ILE.CB	5.0
LEU.CB	LEU.CB	5.0
LYS.NZ	LYS.NZ	5.0
MET.SD	MET.SD	5.0
SER.OG	SER.OG	5.0
THR.OG1	THR.OG1	5.0
TRP.NE1	TRP.NE1	5.0
TYR.OH	TYR.OH	5.0
VAL.CB	VAL.CB	5.0
PHE.PSEU	CG, CD1, CD2, CE1, CE2, CZ	5.0
TYR.PSEU	CG, CD1, CD2, CE1, CE2, CZ	5.0
TRP.PSEU	CD2, CE2, CE3, CZ2, CZ3, CH2	5.0
HIS.PSEU	ND1, NE2	5.0
ASP.PSEU	OD1, OD2, CG	5.0
GLU.PSEU	OE1, OE2, CD	5.0
ASN.PSEU	OD1, ND2, CG	5.0
GLN.PSEU	OE1, NE2, CD	5.0
PRO.PSEU	N, CA, CB, CD, CG	5.0
RES.N	ALA.N, ARG.N, ASN.N, ASP.N, CYS.N, GLN.N, GLU.N, GLY.N, HIS.N, ILE.N, LEU.N, LYS.N, MET.N, PHE.N, SER.N, THR.N, TRP.N, TYR.N, VAL.N	4.0
RES.O	ALA.O, ARG.O, ASN.O, ASP.O, CYS.O, GLN.O, GLU.O, GLY.O, HIS.O, ILE.O, LEU.O, LYS.O, MET.O, PHE.O, PRO.O, SER.O, THR.O, TRP.O, TYR.O, VAL.O	4.0

Table 3.3: 80 physicochemical properties used in FEATURE vector

Atom-based	**Residue-based**	**Secondary structure-based**
ATOM-TYPE-IS-C	RESIDUE_NAME_IS_ALA	SECONDARY_STRUCTURE1_IS_3HELIX
ATOM-TYPE-IS-CT	RESIDUE_NAME_IS_ARG	SECONDARY_STRUCTURE1_IS_4HELIX
ATOM-TYPE-IS-Ca	RESIDUE_NAME_IS_ASN	SECONDARY_STRUCTURE1_IS_5HELIX
ATOM-TYPE-IS-N	RESIDUE_NAME_IS_ASP	SECONDARY_STRUCTURE1_IS_BRIDGE
ATOM-TYPE-IS-N2	RESIDUE_NAME_IS_CYS	SECONDARY_STRUCTURE1_IS_STRAND
ATOM-TYPE-IS-N3	RESIDUE_NAME_IS_GLN	SECONDARY_STRUCTURE1_IS_TURN
ATOM-TYPE-IS-Na	RESIDUE_NAME_IS_GLU	SECONDARY_STRUCTURE1_IS_BEND
ATOM-TYPE-IS-O	RESIDUE_NAME_IS_GLY	SECONDARY_STRUCTURE1_IS_COIL
ATOM-TYPE-IS-O2	RESIDUE_NAME_IS_HIS	SECONDARY_STRUCTURE1_IS_HET
ATOM-TYPE-IS-OH	RESIDUE_NAME_IS_ILE	SECONDARY_STRUCTURE1_IS_UNKNOWN
ATOM-TYPE-IS-S	RESIDUE_NAME_IS_LEU	SECONDARY_STRUCTURE2_IS_HELIX
ATOM-TYPE-IS-SH	RESIDUE_NAME_IS_LYS	SECONDARY_STRUCTURE2_IS_BETA
ATOM-TYPE-IS-OTHER	RESIDUE_NAME_IS_MET	SECONDARY_STRUCTURE2_IS_COIL
ATOM-NAME-IS-ANY	RESIDUE_NAME_IS_PHE	SECONDARY_STRUCTURE2_IS_HET
ATOM-NAME-IS-C	RESIDUE_NAME_IS_PRO	SECONDARY_STRUCTURE2_IS_UNKNOWN
ATOM-NAME-IS-N	RESIDUE_NAME_IS_SER	
ATOM-NAME-IS-O	RESIDUE_NAME_IS_THR	
ATOM-NAME-IS-S	RESIDUE_NAME_IS_TRP	
ATOM-NAME-IS-OTHER	RESIDUE_NAME_IS_TYR	
HYDROXYL	RESIDUE_NAME_IS_VAL	
AMIDE	RESIDUE_NAME_IS_HOH	
AMINE	RESIDUE_NAME_IS_OTHER	
CARBONYL	CLASS1_IS_HYDROPHOBIC	
RING-SYSTEM	CLASS1_IS_CHARGED	
PEPTIDE	CLASS1_IS_POLAR	
	CLASS1_IS_UNKNOWN	
	CLASS2_IS_NONPOLAR	
	CLASS2_IS_POLAR	
	CLASS2_IS_BASIC	
	CLASS2_IS_ACIDIC	
	CLASS2_IS_UNKNOWN	
	PARTIAL-CHARGE	
	VDW-VOLUME	
	CHARGE	
	CHARGE-WITH-HIS	
	NEG-CHARGE	
	POS-CHARGE	
	HYDROPHOBICITY	
	MOBILITY	
	SOLVENT-ACCESSIBILITY	

3.2.1.4 Fragment library construction

A key prerequisite in fragment-based inhibitor discovery is the construction of a suitable fragment library. This library defines the universe of small chemical substructures that can be considered as building blocks for whole compounds.

One approach, exemplified by FragFEATURE, is to derive the fragment library from existing chemical databases. This paradigm screens a large repository (the PubChem database) for small molecules with the desired size (e.g. 3-13 heavy atoms, or MW $\leqslant$ 250 Da) to serve as fragment candidates. This database-driven strategy ensures that fragments inherently obey synthetically accessiblities and cover broad chemical space covered by large-scale databases. However, compiling fragment libraries from a large database suffers from high computational cost when storing and de-duplicating the fetched fragment-like compounds. Moreover, many fragments derived from general-purpose chemical databases may be irrelevant to the specific target families of interest, potentially introducing noise and decreasing predictive signal.

To address these limitations, ChemPLAN-Net introduces a complementary *in silico* fragmentation strategy based on user-defined SMARTS (SMiles ARbitrary Target Specification) rules. This approach allows researchers to specify precise chemical bond patterns to cleave within known ligands, thereby tailoring the fragment library to the structural and mechanistic characteristics of the target class. For example, when focusing on ATP-competitive scaffolds for kinase inhibitors, we may preserve the heteroaromatic "hinge binder" (e.g., purine, pyrimidine, quinazoline)[236] and cleave non-ring single bonds connecting that core to solvent-exposed substituents. While for compounds binding at flat PPIs, we may cut linker motifs (e.g., amide and ether) but keep contiguous hydrophobic rings (e.g., biphenyls) complementary to hot-spot residues such as Trp, Arg, and Tyr[237]. This alternative approach results in different fragment libraries for ATP-competitive inhibitors and PPI modulators.

This rule-based fragmentation approach offers high flexibility in tailoring fragment libraries from general fragment library creation to specialized library containing fragments of desired properties. Therefore, it results in a curated fragment library that balances the completeness of chemical space and relevance to the ligand set under study.

Notably, this fragment curation process is extensible to other existing fragmentation algorithms[238] such as the BRICS method (Breaking of Retrosynthetically Interesting Chemical Substructures)[239] and fragmentation methods[240] for MMPA (Matched Molecular Pair Analysis)[241, 242, 243].

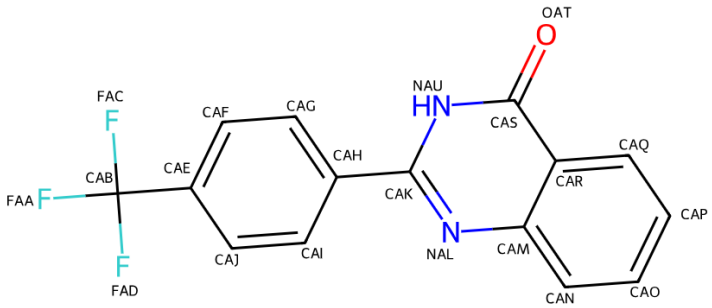


Figure 3.3: An example of ligand (RCSB Ligand code: 06R).

3.2.1.5 Ligand fragmentation via substructure matching

Following the construction of a fragment library, each ligand is scanned for occurrences of these fragments using substructure matching techniques, consistent with the FragFEATURE pipeline. The goal is to identify all subgraphs within a ligand that correspond to predefined fragment structures. This process typically involves an initial fast screening via molecular fingerprints, followed by a more precise graph isomorphism check or SMARTS-based pattern matching.

For every matched fragment, the correspondence between atoms in the fragment and atoms in the ligand is recorded. It is crucial to track the atom-to-atom mapping during this process for associating ligand fragments with specific local environments on the protein. By preserving which specific atoms of the ligand form each fragment, thereby enabling downstream pairing for supervised learning.

While conceptually straightforward, this mapping step is non-trivial in practice. It requires to transfer bond information from the SDF file of a ligand to its corresponding PDB file. The SDF records bond connectivity, while the PDB specifies atom names and spatial positions. Figure 3.3 shows an example ligand (06R), decomposed to fragments listed in Figure 3.4 via substructure matching against a predefined fragment library, with each matching annotated by the atom names from the parent ligand. Since overlapping is allowed during ligand decomposition, one fragment may contain multiple possible mappings. For instance, the toluene fragment in the top row of Figure 3.4 is matched to multiple atom subsets within the same ligand.

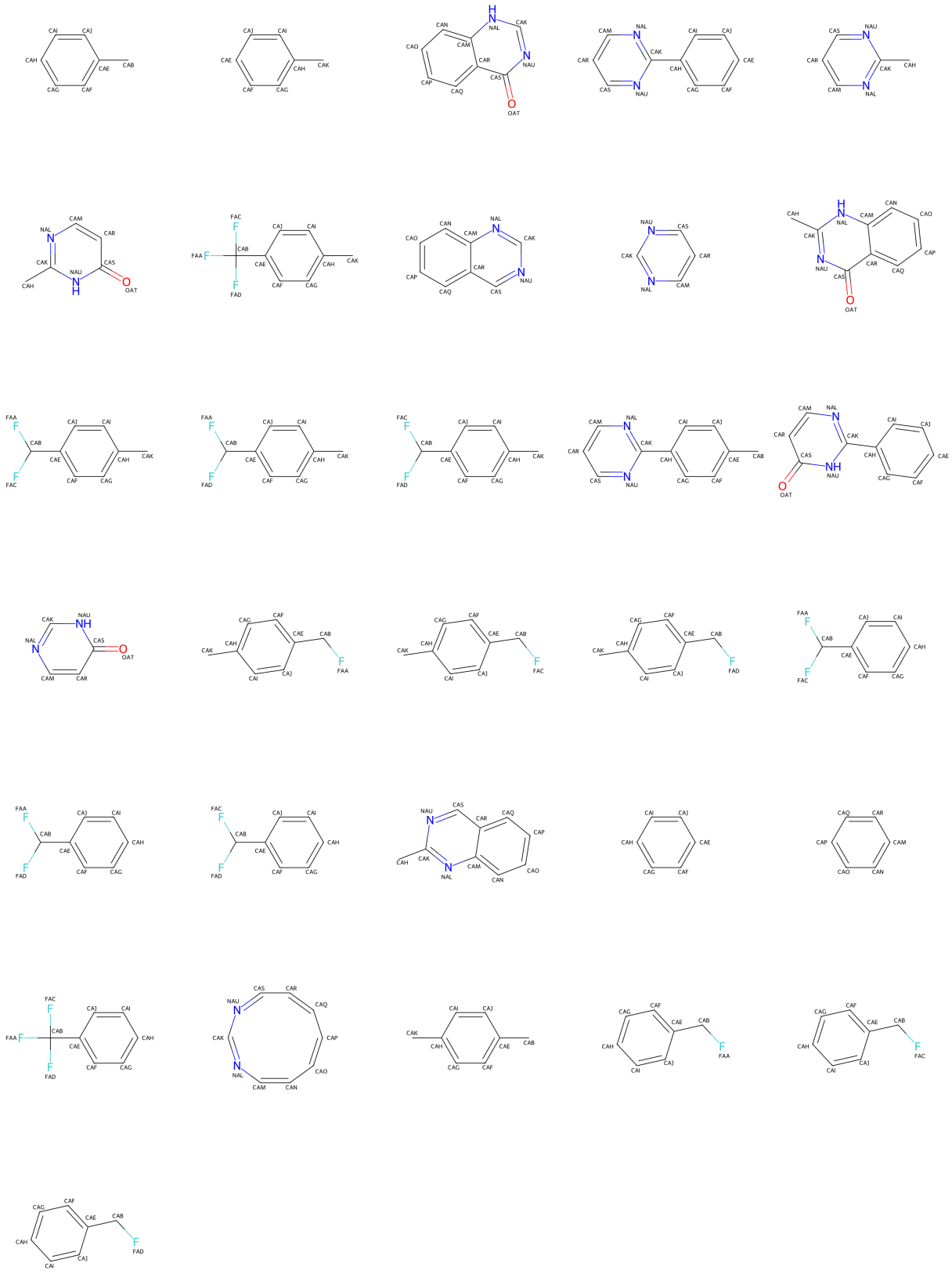


Figure 3.4: An example of ligand (RCSB Ligand code: 06R) decomposed into fragments via substructure matching.

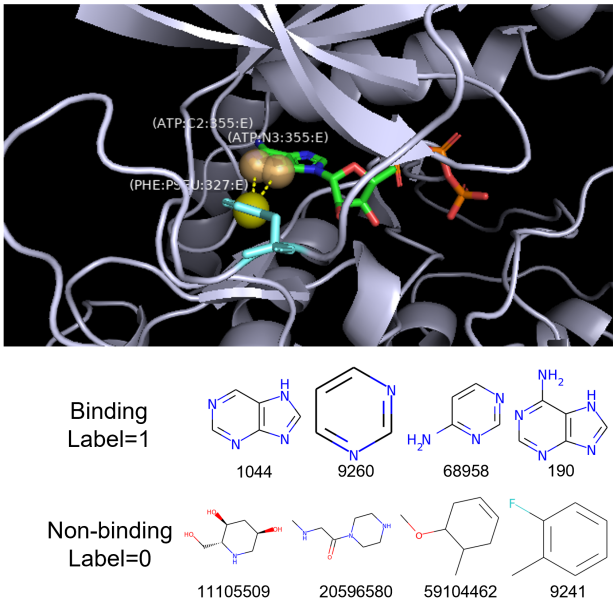


Figure 3.5: Sample non-binding fragments for a given binding site.

3.2.2 Construction of local environment-fragment binding/non-binding pairs

After substructure matching, each protein local environment is associated with a set of binding fragments from the ligand decomposition and atom-to-atom mapping as described in the above Section 3.2.1.5. The dataset is further preprocessed using ball trees to clustering FEATURE vectors and merging their binding fragments.

Unlike the local environment retrieval in FragFEATURE and EnsFragFEATURE, ChemPLAN-Net requires a training dataset with both positive (binding) and negative (non-binding) samples for the binary classification task (Figure 3.5).

To define negative samples, ChemPLAN-Net uses a chemically dissimilar sampling strategy based on fingerprint distance. Morgan fingerprints (radius = 2, 1024 bits) are first generated for every fragment. For each positive (binding) pair, an non-binding fragment is drawn such that its Tanimoto similarity to the binding fragment is below a tunable non-binding fragment separation parameter λ.

Three sampling regimes were evaluated:

- random selection with no similarity constraint where fragments are chosen uniformly from the library
- a strict separation with $\lambda < 0.10$ in which only the most orthogonal chemotypes are

selected

- a relaxed but chemically distinct regime with $\lambda < 0.25$

A systematic benchmark of λ on HIV-1 protease showed that the choice of dissimilarity threshold controls the realism and the difficulty of the negative class. When non-binding fragments were sampled at random, all three tested models (ChemPLAN-Net, SVM and Random Forest) achieved only 50% accuracy no better than guessing. Applying a strict threshold $\lambda < 0.10$ improved classification accuracy to 84.7%, but at the expense of compromised atom coverage in practice, because the decision boundary had been trained on overly simplistic negatives. A relaxed threshold $\lambda < 0.25$ delivered a moderate accuracy of 80.3% but markedly higher recovery of ground-truth fragments. Therefore, this moderate threshold was adopted for all subsequent training and validation. By sampling one dissimilar fragment to form a negative pair for each existing binding pair, the final training set retains one-to-one class balance without overwhelming the learner with redundant negatives.

3.2.3 Neural network architecture

ChemPLAN-Net reformulates protein-fragment prediction as a binary association task by pairing a protein local environment with a candidate ligand fragment and estimating their binding likelihood via a deep residual network.

As illustrated in Figure 3.6, the physicochemical context of a protein binding site is encoded as a 6×80 FEATURE vector which counts 80 atomic, residue and secondary structure properties across six concentric shells. This 6×80-dimensional tensor is processed by a series of sequential ResNeXt[244] blocks, each block comprising grouped 3×3 convolutions, bottleneck 1×1 projections and residual connections[183], to capture higher-order correlations among features in adjacent shells while mitigating gradient vanishing issues. By stacking a series of CNN blocks, the model enables to capture both short- and long-range dependencies from inner shell to the outer shell of the FEATURE vector.

The latent tensor describing protein environment from the final ResNeXt block is flattened and concatenated with a 1024-bit Morgan fingerprint (radius 2) representing the topological environment of the candidate fragment. This yields a fused representation that captures binding site–fragment complementarity. A two-layer multilayer perceptron with batch normalization and dropout maps the fused vector to the predicted binding probability $\in [0, 1]$. During training, experimentally observed binding site–fragment pairs are labelled positive, whereas non-binding fragments are synthetically generated by Tanimoto-dissimilarity sampling. The model is trained using the binary cross-entropy loss function.

By focusing on local environment descriptors rather than global sequences or structures, ChemPLAN-Net generalizes to novel proteins that share similar binding-site environments due to evolutionary structural conservation. Likewise, using Morgan fingerprints enables the model to recognize novel fragments outside the training library that may share similar motifs or scaffolds with known binders in a leave-5%-out experiment. More generally, the continuous embedding learned for molecular fingerprints allows ChemPLAN-Net to infer previously unseen chemistry, a capability unavailable to either FragFEATURE or its ensemble extension.

The ChemPLAN-Net model was initially trained and validated using HIV-1 protease as a benchmark system. During training, HIV-1 protease and structures with 40% sequence identity were held out for test. The remaining dataset was further split to training and validation randomly using a 98 : 2 split ratio.

The HIV-1 protease task provided a well-characterized platform to optimize hyperparameters, assess fragment coverage metrics, and validate the Tanimoto-based negative sampling regime. To assess chemical relevance, fragment predictions were mapped back to native ligands, and atom coverage was computed as a measure of recall at the structural level for native ligands. Performance on this system where ChemPLAN-Net achieved an atom coverage ratio of $63\% \pm 19\%$ (Figure 3.7A) of native inhibitors, outperforming its fragment docking baseline ($\sim 21\% \pm 22\%$) and Random Forest (RF) baseline ($\sim 59\% \pm 22\%$)[4]. This served as a reference for extrapolating to unseen proteases, such as SARS-CoV-2 M^{pro}.

3.3 Results

3.3.1 SARS-CoV-2 M^{pro} inhibitors

The SARS-CoV-2 M^{pro} is a critical antiviral drug target due to its essential role in processing viral polyproteins. The following results are collected and adapted from ChemPLAN-Net published results[4] and the PhD thesis of Dr. Michael Suarez[245].

ChemPLAN-Net trained on the entire protease dataset, validated in Section 3.2.3, was applied to 100 co-crystal structures of SARS-CoV-2 M^{pro} and its inhibitors. After redundancy reduction, 228 unique local environments were identified and screened against a predefined library of approximately 60k fragments. Fragments receiving a predicted binding probability above 0.97 were retained, resulting in 143 distinct fragment predictions across these environments. Clustering analysis using Tanimoto similarity further revealed that ChemPLAN-Net produced 16 chemically distinct fragment clusters (vs. 15 for RF), indicating enhanced chemical diversity. While 44 of the 143 predicted fragments

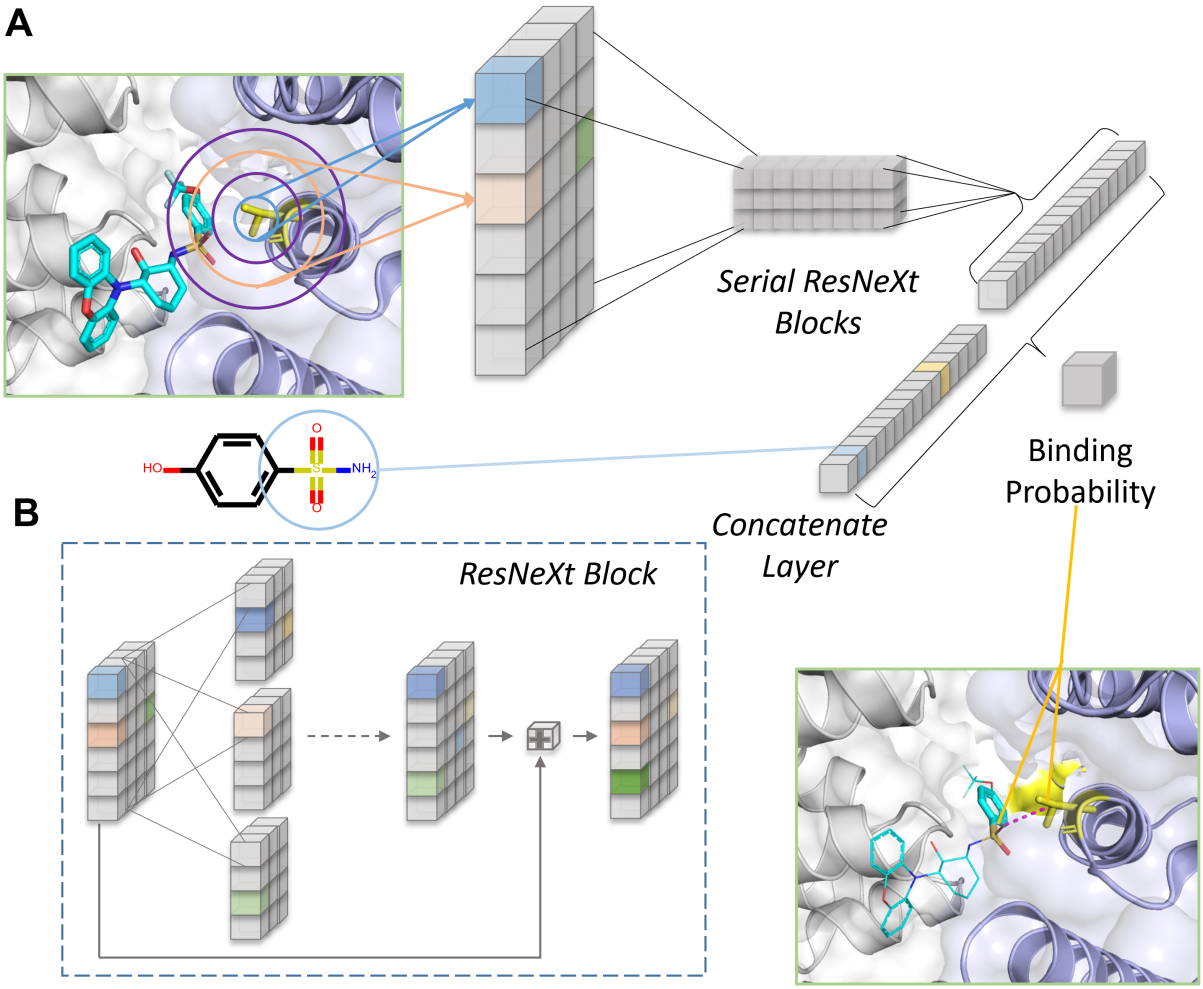


Figure 3.6: ChemPLAN-Net architecture. (A) FEATURE vectors containing binding site information are processed using ResNeXt blocks, and then concatenate with fragment fingerprints to output a binding probability. (B) Implementation of ResNeXt block.

matched substructures in native ligands, yielding a raw precision of 32.8%, this value increased to 81.3% when evaluated at the cluster level. Furthermore, as shown in Figure 3.7B, ChemPLAN-Net achieved a mean atom coverage ratio of 55% $\pm$ 27% for 105 SARS-CoV-2 M^{pro} ligands. These results demonstrate that ChemPLAN-Net captures meaningful chemotype-level binding preferences and achieve transferability to other protein system within the same protein family.

In summary, ChemPLAN-Net trained on protease effectively generalized from HIV-1 protease to SARS-CoV-2 M^{pro}. Its ability to recover diverse and chemically relevant fragments at high precision underscores its potential as a deep learning–based engine for fragment prioritization in antiviral drug discovery pipelines.

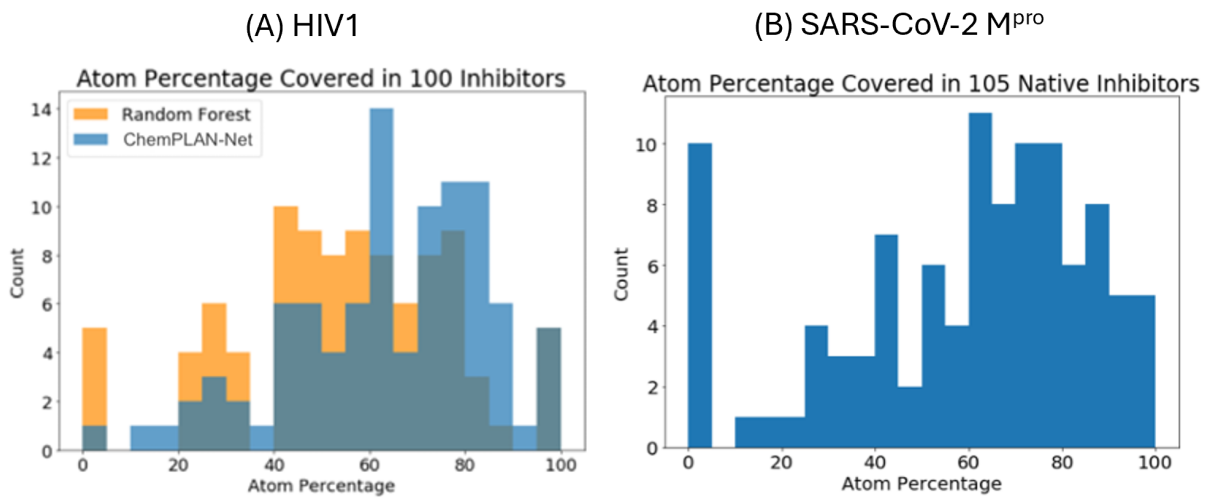


Figure 3.7: Atom coverage distribution of ligands in cocrystal structures of interest. (A) Atom coverage distribution of 100 native ligands in HIV-1 cocrystal structures. Reproduction from ChemPLAN-Net[4] Figure 3b under a CC-BY-NC 4.0 International license. (B) Atom coverage distribution of 105 native ligands in Sars-CoV-2 M^{pro} cocrystal structures. Reproduction from Dr. Michael Suarez's PhD thesis Figure 23b under a CC BY-NC-ND 3.0 license

3.3.2 A systematic evaluation on proteases, kinases and phosphatases

To evaluate the generalization capacity of ChemPLAN-Net across diverse protein families, a systematic assessment was conducted on proteases, kinases, and phosphatases. For each family, datasets were curated using the automated pipeline described in Section 3.2.1, beginning with PDB entries filtered by their corresponding Enzyme Commission (EC) numbers. Model training followed the same fragment pairing and architecture setup outlined in Sections 3.2.2 and 3.2.3.

Representative proteins were selected for evaluation in each family: HIV-1 protease for proteases, CDK2 for kinases, and PTP1B for phosphatases. To ensure a fair and rigorous evaluation, all structures sharing greater than 40% sequence identity with the test target were excluded from the training set. The remaining structures were randomly partitioned into training and validation sets. All these split strategies followed description in Section 3.2.3.

Training accuracy curves are shown in the top panel of Figure 3.8. ChemPLAN-Net exhibited stable and consistent convergence across all three protein families. To assess chemical relevance, atom coverage were calculated by mapping predicted fragments back to native ligands. These distributions are presented in the bottom panel of Figure 3.8.

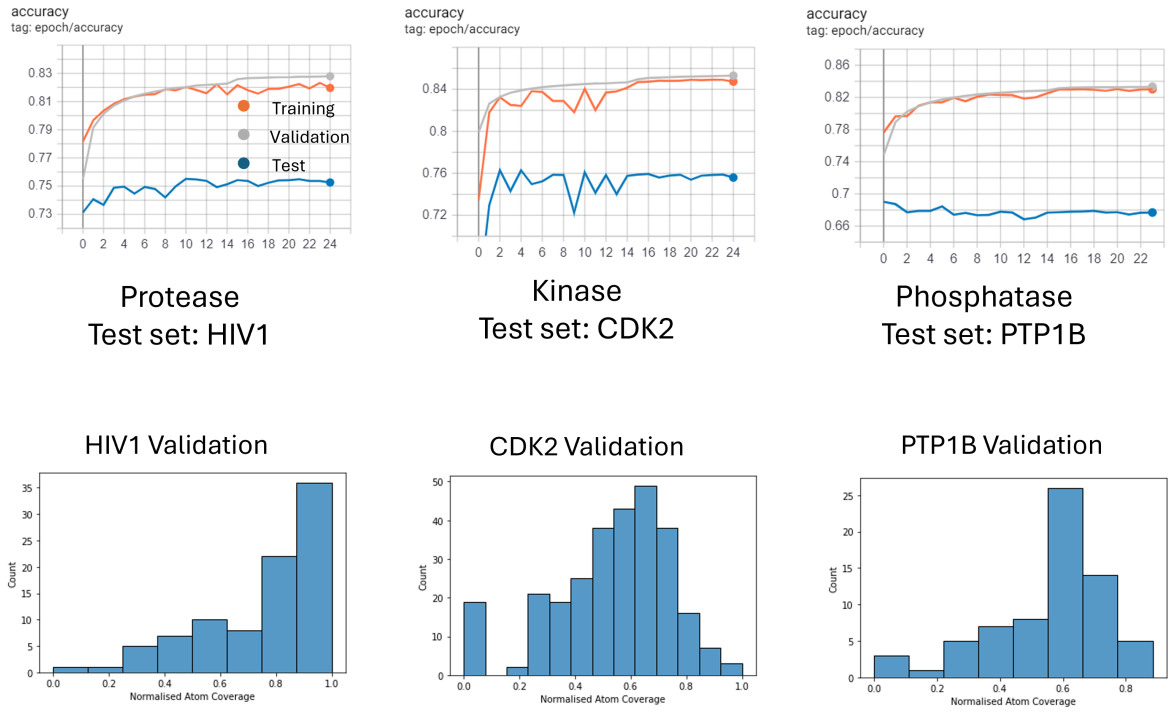


Figure 3.8: Systematic evaluation of ChemPLAN-Net across various protein families: proteases validated using HIV-1, kinases validated using CDK2, and phosphatase validated using PTP1B. **Top panel**: Training curves. **Bottom panel**: The distributions of ligand atom coverage on three different proteins, one from each protein family.

Notably, the model achieved the highest atom coverage performance on HIV-1 protease ligands. One possible explanation is the symmetric binding mode of ligands within the homodimeric HIV-1 protease. In such systems, a single correctly predicted substructure is often redundantly mapped to both symmetric subpockets, inflating the coverage metric. In contrast, ligands in CDK2, PTP1B, and SARS-CoV-2 M^{pro} (from Figure 3.7B) bind to asymmetric, monomeric sites, resulting in more conservative coverage scores. Nevertheless, consistent atom coverage distributions across CDK2 and PTP1B support the model's applicability beyond proteases.

It is important to note that results for HIV-1 protease reported here may differ slightly from those shown in Figure 3.7A, due to differences in dataset construction. The systematic evaluation in Figure 3.8 was performed using a unified curation pipeline across all families following descriptions in Section 3.2.1, ensuring comparability but introducing minor variations in input data.

Another important observation from the top panel of Figure 3.8 is the noticeable perfor-

mance gap between training/validation accuracy and test accuracy across all three protein families. This discrepancy is attributable to differences in the data splitting strategy. While training and validation sets were generated via random splits, allowing for some structural similarity between the two, the test set was constructed using a strict hold-out protocol, excluding any proteins with greater than 40% sequence identity to the test target. As a result, the test set poses a significantly more challenging generalization task, better reflecting real-world scenarios where novel targets differ substantially from training examples. This observation further underscores the importance of rigorous evaluation design when assessing model generalizability.

3.3.3 Molecular glues

To evaluate the potential of ChemPLAN-Net beyond prediction of inhibitive fragments for individual proteins, the model was trained on a curated dataset of PDB entries filtered using the pipeline described in Section 3.2.1.1. This dataset was specifically assembled to model ligand interactions at protein–protein interaction (PPI) interfaces, with the ultimate goal of identifying novel fragments that could be further developed into molecular glues.

The training follows a similar training/validation/test split and the same training protocol. The trained model was evaluated on the ternary-complex structure of the retinoic-acid-receptor/ nuclear-corepressor (RAR/NCoR) and CDK12/DDB1 protein–protein interfaces. The fragments assigned a binding probability greater than 0.90 were considered as high confidence and mapped onto the molecular-glue ligands.

Under these conditions the network reconstructed almost the entire glue chemotype for RAR/NCoR, covering 93.55% of the heavy atoms (Figure 3.9A). Performance on the mechanistically distinct CDK12/DDB1 interface was also robust, with high-probability predictions accounting for 65.52% of heavy atoms (Figure 3.9B).

While promising, the results do not readily generalize to other systems, as the application of ChemPLAN-Net to protein–protein interaction (PPI) modulators reveals several intrinsic challenges. A fundamental distinction between the curated PPI dataset and those derived from single protein families (e.g., proteases or kinases) lies in the structural and functional heterogeneity of their respective local environments. In enzyme families, ligand binding typically occurs within well-defined and evolutionarily conserved active sites, characterized by recurring interaction motifs and chemically coherent binding environments. In contrast, PPI interfaces are often shallow, topologically diverse, and compositionally variable. This heterogeneity disrupts the formation of consistent interaction patterns within the training dataset, complicating the model's ability to learn transferable local environment-

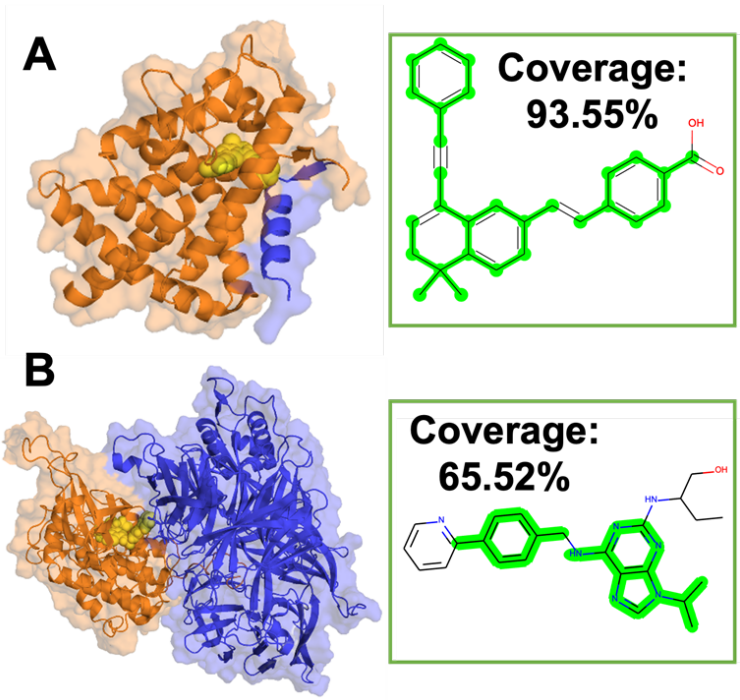


Figure 3.9: Atom coverage ratio of ChemPLAN-Net predicted fragments on molecular glues binding to (A) RAR/NCoR (PDB id: 3KMZ), and (B) CDK12/DDB1 (PDB id: 6TD3).

fragment associations. As a result, predictive performance on PPI modulators is less stable and more difficult to generalize compared to well-characterized single-target systems.

Second, due to the limited number of experimentally validated and annotated molecular glue ternary complexes available in the PDB, the training dataset was necessarily expanded to include both PPI inhibitors and potential stabilizers. While this broadened the available data, it also introduces mechanistic ambiguity that may impair the model's ability to learn the specific binding features associated with true PPI-enhancing molecular glues. The mixture of inhibitory and stabilizing modes of action in training may hinder the precision of fragment predictions when applied to genuine glue candidates. Future work may incorporate MD simulations to sample interface conformational ensembles, augmenting the training dataset of true molecular glues.

3.3.4 Discussions

While ChemPLAN-Net exhibits strong predictive performance on structurally conserved protein families such as proteases and kinases, a closer examination of its training data distribution reveals notable sources of imbalance that may hinder its generalizability, especially to structurally heterogeneous systems such as protein–protein interfaces (PPIs).

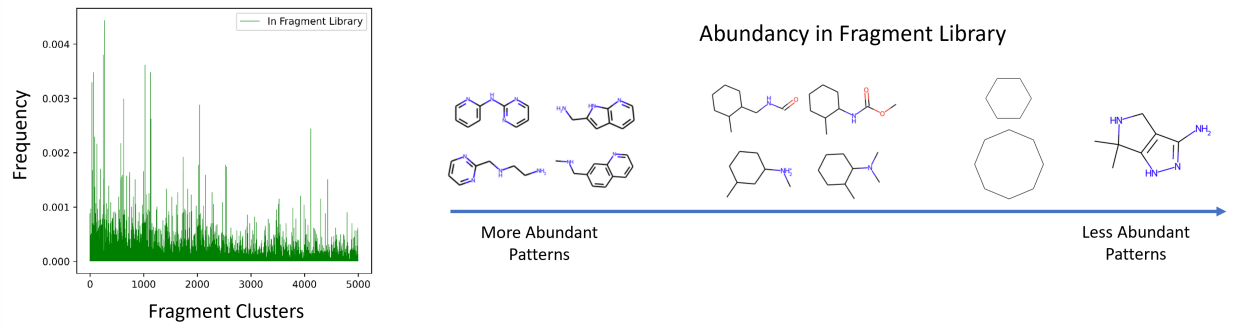


Figure 3.10: Fragment redundancy in the pre-defined fragment library. **Left panel**: Fragments were clustered into 5,000 groups using Agglomerative Clustering on Tanimoto similarity. **Right panel**: Illustration of more versus less abundant chemical motifs across the fragment space.

As described in Section 3.2.2, binding fragments corresponding to similar protein local environments are merged to reduce redundancy. The remaining, non-redundant environments, referred to as "non-redundant FEATURE vectors" later, form the basis of the training dataset. However, this redundancy reduction does not resolve a more subtle issue that the unequal number of binding (and non-binding) fragments associated with each FEATURE vector. Some environments are associated with hundreds of fragments, while others with very few. As shown in the left panel of Figure 3.11, this disparity results in certain FEATURE vectors being sampled more frequently during training, thereby increasing their influence on model optimization.

The original implementation of ChemPLAN-Net does not address this form of imbalance through dynamic sampling. Ideally, during each training epoch, a resampling strategy such as selecting one binding and one non-binding fragment per non-redundant FEATURE vector would prevent the model from being disproportionately shaped by high-fragment-density environments.

Moreover, an additional and compounding source of imbalance arises from the fragment library itself, which is not trivial to avoid and requires deliberate design choices during library construction, dataset curation and training. The pre-defined fragment library used for substructure matching contains roughly 60k fragments, many of which are structurally redundant or highly similar. By clustering these fragments into 5,000 groups using Agglomerative Clustering and Tanimoto distance, the imbalance in different fragment clusters becomes evident (Figure 3.10). ChemPLAN-Net does not account for this redundancy during dataset curation as well as in training.

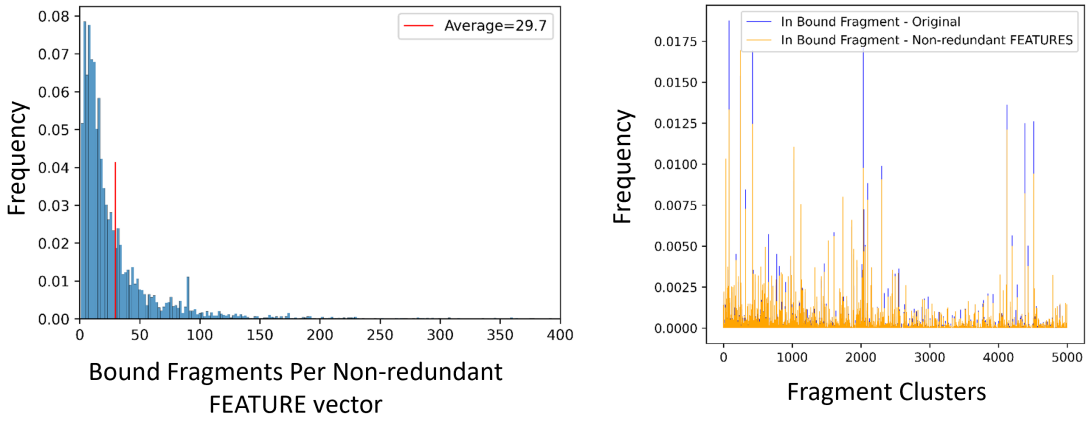


Figure 3.11: Twofold imbalance in the training data. **Left panel**: Distribution of the number of binding fragments per non-redundant FEATURE vector. **Right panel**: Distribution of fragment usage across the dataset, showing overrepresentation of certain fragments due to redundancy in the fragment library and the ligand decomposition protocol.

This problem is further exacerbated by the use of overlapping-allowed substructure matching during ligand decomposition. This approach may disproportionately favor frequent substructures from chemically prevalent clusters and artificially inflates their presence among identified binding fragments, other than reflecting the true prevalence caused by binding.

The coupled effect lead to the final fragment-level imbalance in the curated dataset, illustrated in the right panel of Figure 3.11, where a small subset of fragment clusters dominates the training data, with real signals mixed with library redundancy and curation artifacts. As a result, fragment clusters with high internal redundancy become overrepresented in the input space, as structurally similar fragments produce highly correlated fingerprint representations. This redundancy skews the training distribution, increasing the likelihood that the model will favor these chemically prevalent fragments regardless of their true binding compatibility.

These two sources of imbalance, from FEATURE vectors and from fragments, may significantly undermine the intended function of ChemPLAN-Net. While the model is designed to learn the nuanced interplay between protein environments and fragment chemistry, in the worst case scenario, it may instead learn the fragment frequency bias, thereby diminishing its ability to generalize beyond overrepresented patterns in the training data.

This bias may remain obscured in single-family models where fragment binding envi-

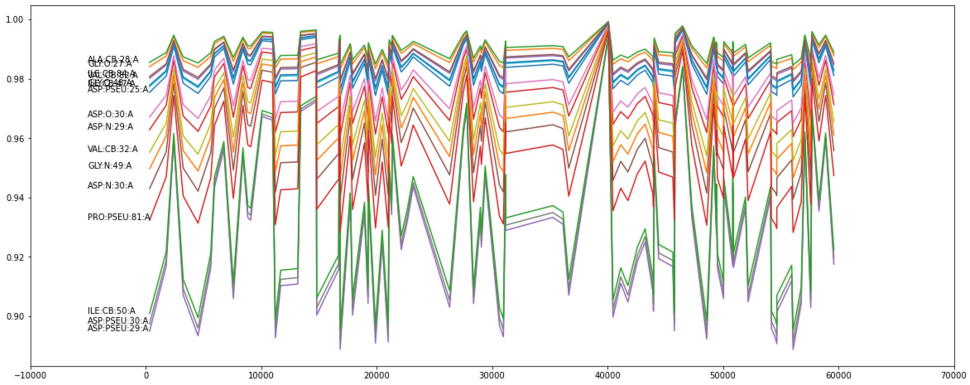


Figure 3.12: Fragment rankings are not sensitive to FEATURE vector variation. Each zigzag line represents the predicted binding probabilities of top-ranked fragments for each functional center in the binding pocket of HIV-1 protease (PDB ID: 4FE6). Despite the difference in predicted probabilities, the rankings hardly change.

ronments are relatively conserved using evaluation metrics such as accuracy and recall, while becomes more problematic in structurally diverse systems such as PPIs, where interaction specificity is harder to capture and statistical regularities are weaker.

However, the presence of such bias can be revealed through sensitivity analyses. For instance, fragment ranking results across different functional centers within the HIV-1 protease binding pocket (PDB ID: 4FE6) show minimal variation, suggesting that ChemPLAN-Net's predictions are insufficiently sensitive to local changes in the FEATURE vector. As illustrated in Figure 3.12, each zigzag line traces the binding probability of a top-ranked fragment across various local environments within the same pocket. The limited fluctuation in scores indicates that fragment predictions are largely invariant to subtle environmental differences, undermining the model's capacity to discriminate between fine-grained binding site features.

3.4 Applications in virtual synthesis and lead compounds identification for SARS-CoV-2 M^{pro}

To discover novel full-size compounds built from ChemPLAN-Net fragment predictions, a virtual synthesis pipeline (Figure 3.13) was developed to construct a compound library of acylhydrazone derivatives suitable for fragment-based screening and downstream lead optimization, using the aldehyde-derived hydrazones synthesis[246]. The objective was to rapidly generate, filter, and prioritize compounds with high predicted binding compatibility to SARS-CoV-2 main protease (M^{pro}) fragments identified by ChemPLAN-Net which can

be quickly synthesized and tested in the laboratory.

3.4.1 Fragments preparation and virtual synthesis

The virtual synthesis contains three steps. In the initial stage, 3,581 commercially accessible carboxylic acids were retrieved from the ZINC database through a sub-structure query enforcing the canonical carboxylate SMARTS pattern "CX3[OX2H1]". Each carboxylic acid was converted *in silico* into its corresponding hydrazide via a single-step amidation with hydrazine, a transformation that is routinely achieved experimentally under mild peptide-coupling conditions and is therefore considered practically feasible for later hit resynthesis. Parallel to this, 380 structurally diverse aldehydes that previously procured and characterized by Prof. Weiping Tang's laboratory[247].

In the second stage, every hydrazide was virtually reacted with every aldehyde using the reaction SMARTS: `[CH1:2]=[O:2].[NH2:4][NH1:5]>>[CH1:2]=[N:4][NH1:5]`, resulting in a theoretical combinatorial space of 1,368,000 compounds. Removal of duplicate structures that originated from symmetry or tautomeric redundancy yielded 1,360,780 unique acylhydrazone products, underscoring the vast chemical space accessible through FBDD in conjunction with combinatorial chemistry.

In the last stage of the virtual synthesis, the corresponding amide isosteres will be synthesized because acylhydrazones display hydrolytic instability under physiological conditions.

3.4.2 Filtering using drug-like properties

The enumerated library was subsequently filtered with using Lipinski's Rule of Five to enhance the likelihood of obtaining drug-like candidates. Molecules exceeding 600 Da, or logP values outside the 2-5 (estimated by RDkit's (Descriptors.MolLogP), were excluded. Approximately 49% of the candidates were removed in this step, resulting in a final virtual screening library of 689,765 acylhydrazone derivatives that balance diversity with drug-likeness.

3.4.3 Filtering using atom coverage

To identify candidate compounds aligned with ChemPLAN-Net predictions, 90 fragments with predicted binding probabilities exceeding 0.99 from the ChemPLAN-Net model trained on protease systems, were selected and mapped onto the virtual synthesized library. The binding pocket was defined using proximal residues to the native ligand UJ4

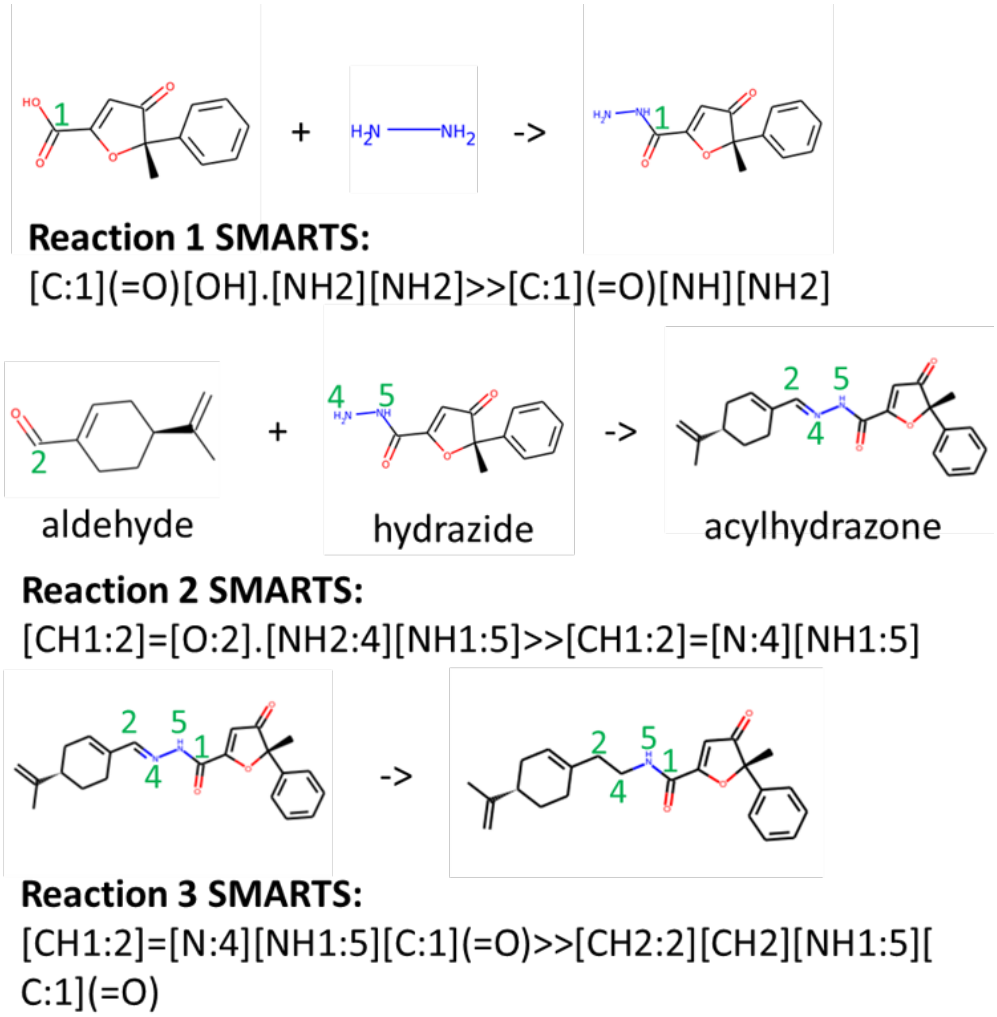


Figure 3.13: A three-step SMARTS-based virtual synthesis between 380 aldehyde molecules and 3581 commercially purchasable carboxylic acids fetched from Zinc.

in SARS-CoV-2 M^{pro} (PDB ID: 5RH9), with docking coordinates centered at $[9, 2, 21]$ and a box size of $14 \times 14 \times 14$ Å. Atom coverage was calculated for each compound based on the number of heavy atoms overlapped with the high-confidence predicted fragments.

As shown in Figure 3.14, the coverage distribution reveals that 197 compounds in the final library achieve complete (100%) atom coverage by 90 predicted fragments. Further analysis (Figure 3.15) identifies several key aldehyde and acid building blocks that recurrently contribute to high-coverage compounds, supporting the fragment-driven rationale of the virtual synthesis strategy.

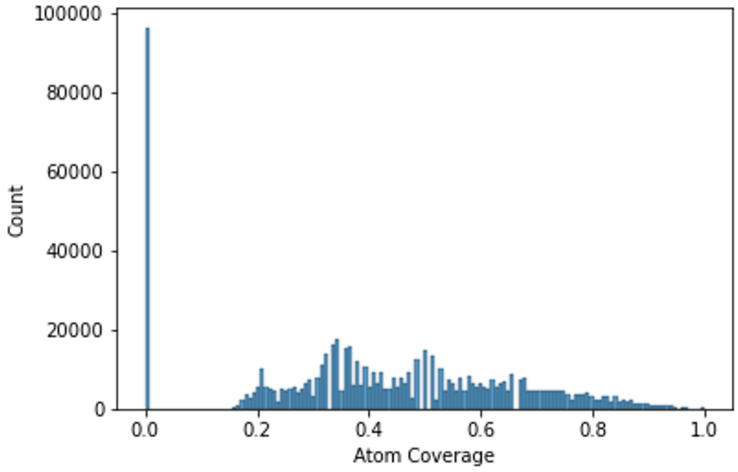


Figure 3.14: Atom coverage distribution of ~690k virtually synthesized compounds against 90 predicted fragments with probabilities > 0.99 for the Sars-Cov-2 Main Protease. The binding pocket is defined by the proximal residues around the native ligand UJ4 in PDB ID: 5RH9. The docking box is centered at $[9, 2, 21]$ with the size of 14Å $\times$ 14Å $\times$ 14Å. Among the synthesized compound library, 197 compounds have 100% atom coverage.

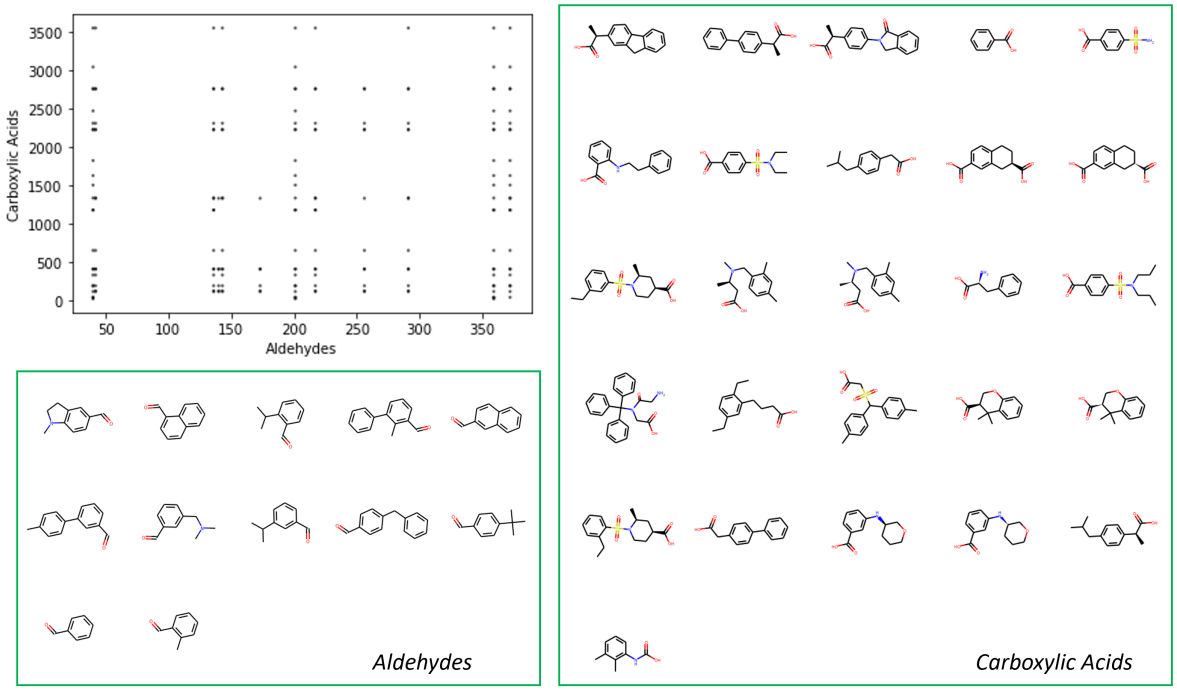


Figure 3.15: Aldehyde and carboxylic acid building blocks contributing to the $>$ 99% coverage compounds

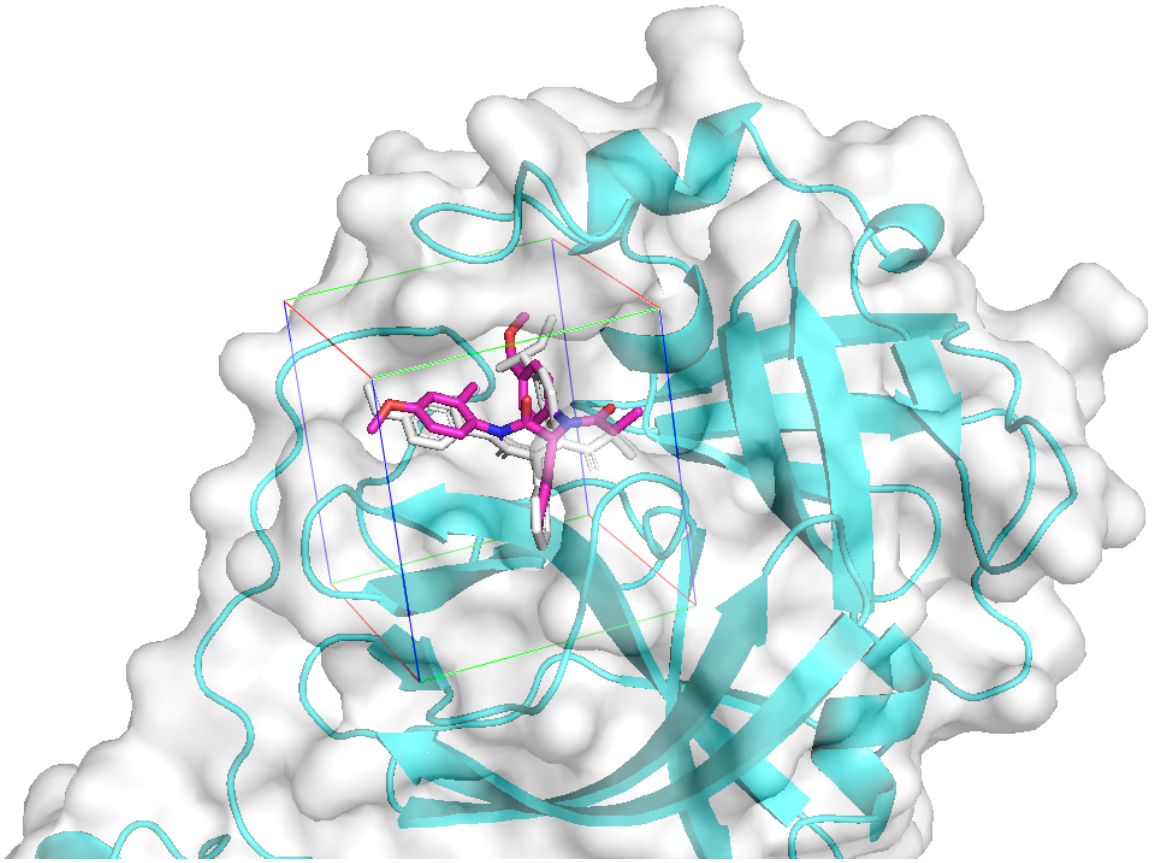


Figure 3.16: Redocked result of PDB ID: 5RH9 with Vina affinity -7.0 kcal/mol. The SARS-CoV-2 Mpro is colored in cyan, the native pose of ligand is colored by white sticks. The redocked pose is colored in magenta sticks.

3.4.4 Structual evaluation using docking

Figure 3.16 shows an example of successful redocking (PDB ID: 5RH9) with a predicted AutoDock Vina affinity of –7.0 kcal/mol. This serves as a baseline reference for the virtual screening of top-ranked compounds, as we want to find novel compounds with stronger affinities than the native ligand.

A total of 197 top-ranked compounds, each achieving 100% atom coverage with respect to high-confidence ChemPLAN-Net–predicted fragments, were docked into the M^{pro} pocket, with the poses and docking affinities of the best compounds shown in Figure 3.17. Interestingly, even though the four visualized compounds all achieved higher docking affinity than the native ligand, they were found to occupy only partial pockets, indicating sub-optimal binding and highlighting the need for further structural optimization.

To evaluate whether atom coverage is a reliable indicator to select compound candidates,

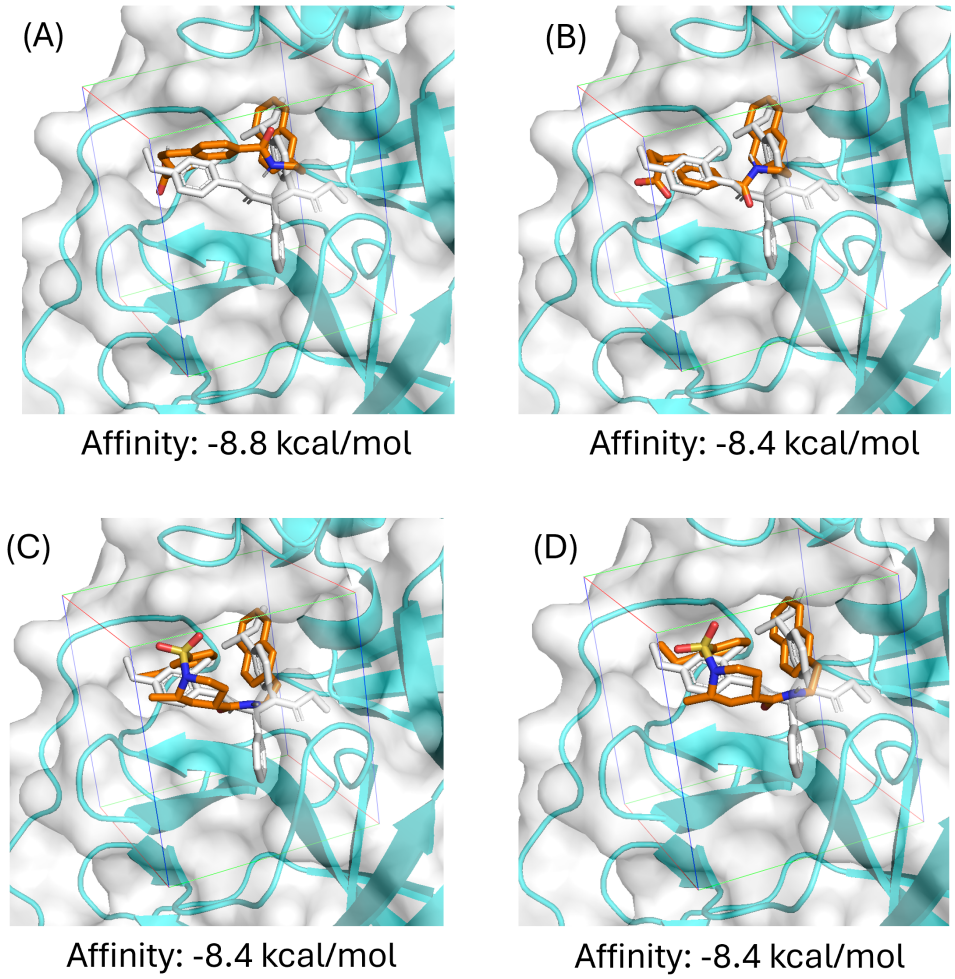


Figure 3.17: Four examples with the best AutoDock Vina affinities from 197 top-ranked compounds based on atom coverage. Orange sticks represents the docked compound poses, overlaid on white sticks that show the baseline native ligand pose.

a comparative docking analysis was conducted. As shown in Figure 3.19, compounds with high atom coverage (100%) surprisingly exhibited weaker average docking affinities than those with lower coverage (10–40%). Several low-coverage compounds with strong docking affinities and favorable binding poses, as visualized in Figure 3.18.

3.4.5 Discussions

This discrepancy may indicate a key limitation of ChemPLAN-Net. While fragment predictions are highly informative for matching substructures to protein local environments, they do not account for global conformational strain, steric clashes, or entropic penalties that

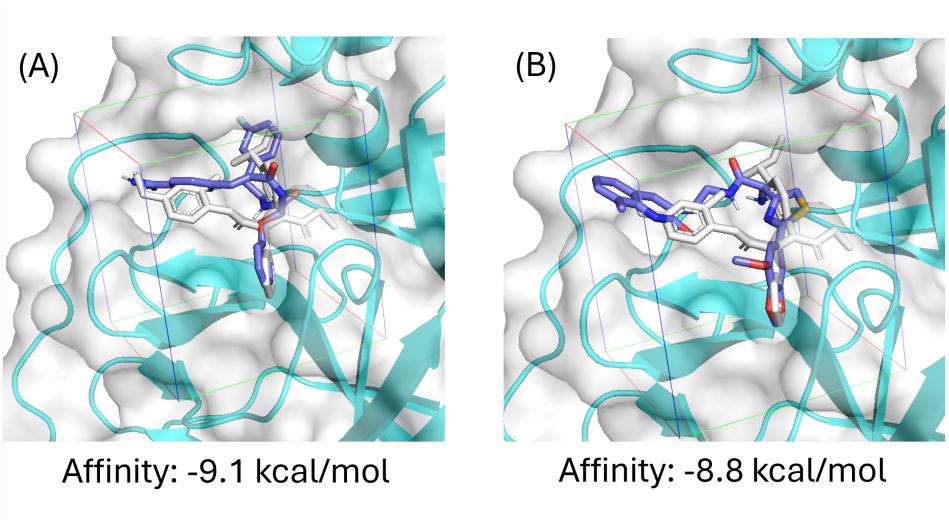


Figure 3.18: Examples with the best AutoDock Vina affinities from compounds with atom coverage in the range of 10-40%. Purple sticks represents the docked compound poses, overlaid on white sticks that show the baseline native ligand pose.

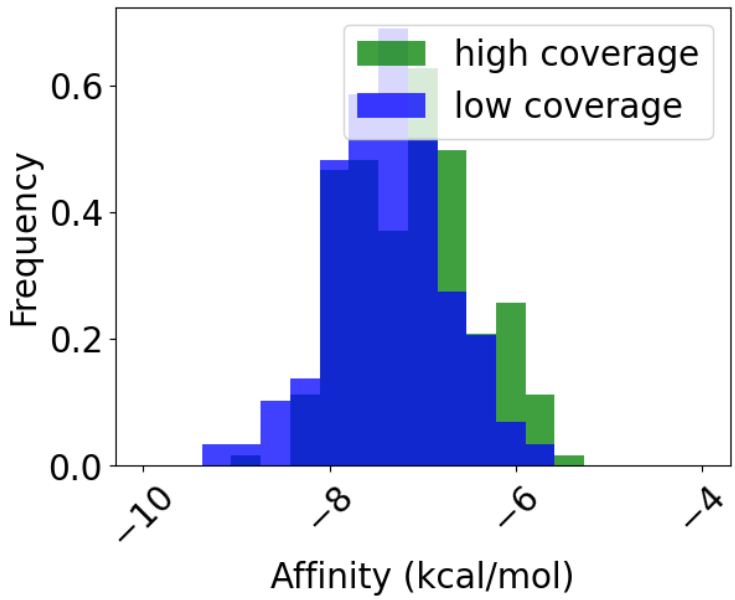


Figure 3.19: Compare Vina affinity of high coverage (100%) compounds with those of low coverage (10%-40%).

may arise at the whole-ligand level. Consequently, high atom coverage does not necessarily translate into optimal binding affinity or pocket fit even though it is a useful for fragment recall when evaluating the performance of trained ChemPLAN-Net. These findings suggest that fragment orientation and spatial arrangement must be incorporated explicitly into the fragment-to-ligand design pipeline.

To address this gap, downstream processes such as fragment linking, merging, and growing, particularly when informed by spatial constraints, can help refine fragment-based leads into viable drug candidates.

3.5 Applications in fragment linking

Integrating ChemPLAN-Net predictions with structure-aware ligand assembly strategies thus represents a promising future direction to enhance both the quality and relevance of virtual screening hits. We applied a structure-aware generative linker design strategy using DeLinker[79]. DeLinker is a graph-based deep generative model trained to generate chemical linkers between pairs of fragments given their 3D conformations, starting atoms, and desired linker length constraints.

To prepare the initial fragment pair and its conformation, the binding pocket of SARS-CoV-2 M^{pro} was first analyzed using FPocket[20], a geometry-based cavity detection tool. FPocket identifies concave regions within the protein structure by computing alpha spheres which indicate potentially ligandable concaves. The set of alpha spheres identified within the binding site of PDB structure 5RH9 (Figure 3.20) was then clustered using the DBSCAN algorithm to partition them into spatially distinct regions. This process yielded four major spatial clusters. As illustrated in Figure 3.20 right panel, one of these centers was located outside the native ligand occupied region and the rest three centers located inside the native ligand occupied region. The remaining three clusters were retained and designated as Center 1, Center 2, and Center 3.

Each center was used to define a cubic docking box of 10Å per side, located at the clustering centers. These defined volumes provided localized environments for fragment docking, ensuring focused sampling within structurally distinct subregions of the active site. Top fragments predicted by ChemPLAN-Net were individually docked to Center 2 and Center 3 with docking centers and boxes described above using LeDock[248].

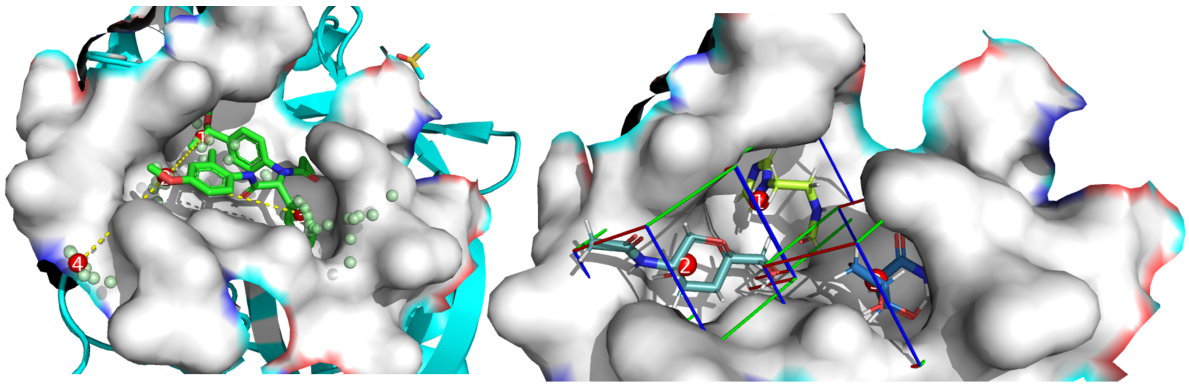


Figure 3.20: An example of fragment docking to subpockets identified by DBSCAN clustering alpha spheres from fpocket results. The centers of four identified subpockets are represented as red dots, and alpha spheres from fpocket are represented as cyan blue dots.

3.5.1 Setup and linker generation

Conformations with proper orientation and high LeDock affinity were hand-picked (Figure 3.21) as initial setup for DeLinker. Specifically, the atoms to link are highlighted using the star sign in Figure 3.22. The geometric constraints for linker design were set as follows.

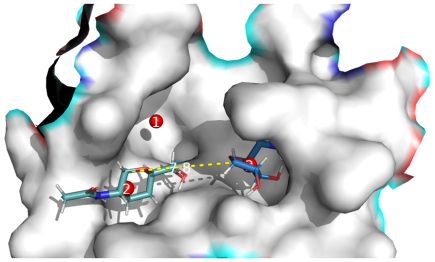
- Euclidean distance between starting atoms: 7.5Å
- Angle between exit vectors: 0.54 radians
- Desired linker length: 7, 8, or 9 heavy atoms

For each of the three linker lengths, 10 conditional generations were performed, yielding a total of 30 full compounds.

3.5.2 2D filtering and deduplication

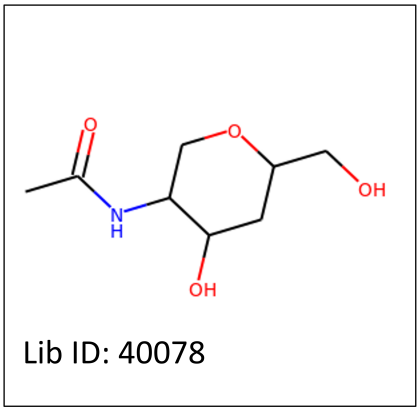
Out of 30 generated compounds, 24 were chemically valid based on RDKit's molecule sanitization routines. These were further filtered through a 2D property pipeline described in DeLinker including:

- Synthetic Accessibility (SA) filter
- Aromatic ring constraint filter
- PAINS (Pan Assay Interference Compounds) filter



Center 2

Fragment Index	Score (kcal/mol)
2435	-3.46
40078	-4.04



Center 3

Fragment Index	Score (kcal/mol)
2435	-4.42
10914	-4.38
38691	-4.22
40078	-4.08
57092	-4.60

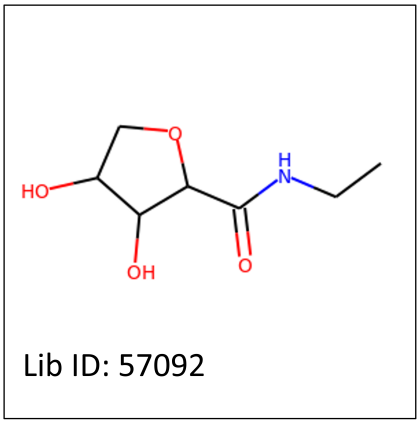


Figure 3.21: Fragment conformation setup for DeLinker prepared from high predicted fragments with good LeDock binding affinities and proper orientation. The dashed yellow line in the PyMol plot connects atoms to link.

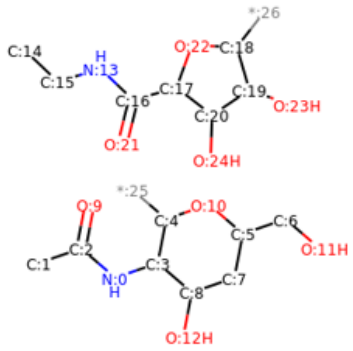


Figure 3.22: Atoms to link (labeled as star) on each fragment are selected based on their orientation. This is a 2D molecular graph generated from SMILES of the fragmented compound.

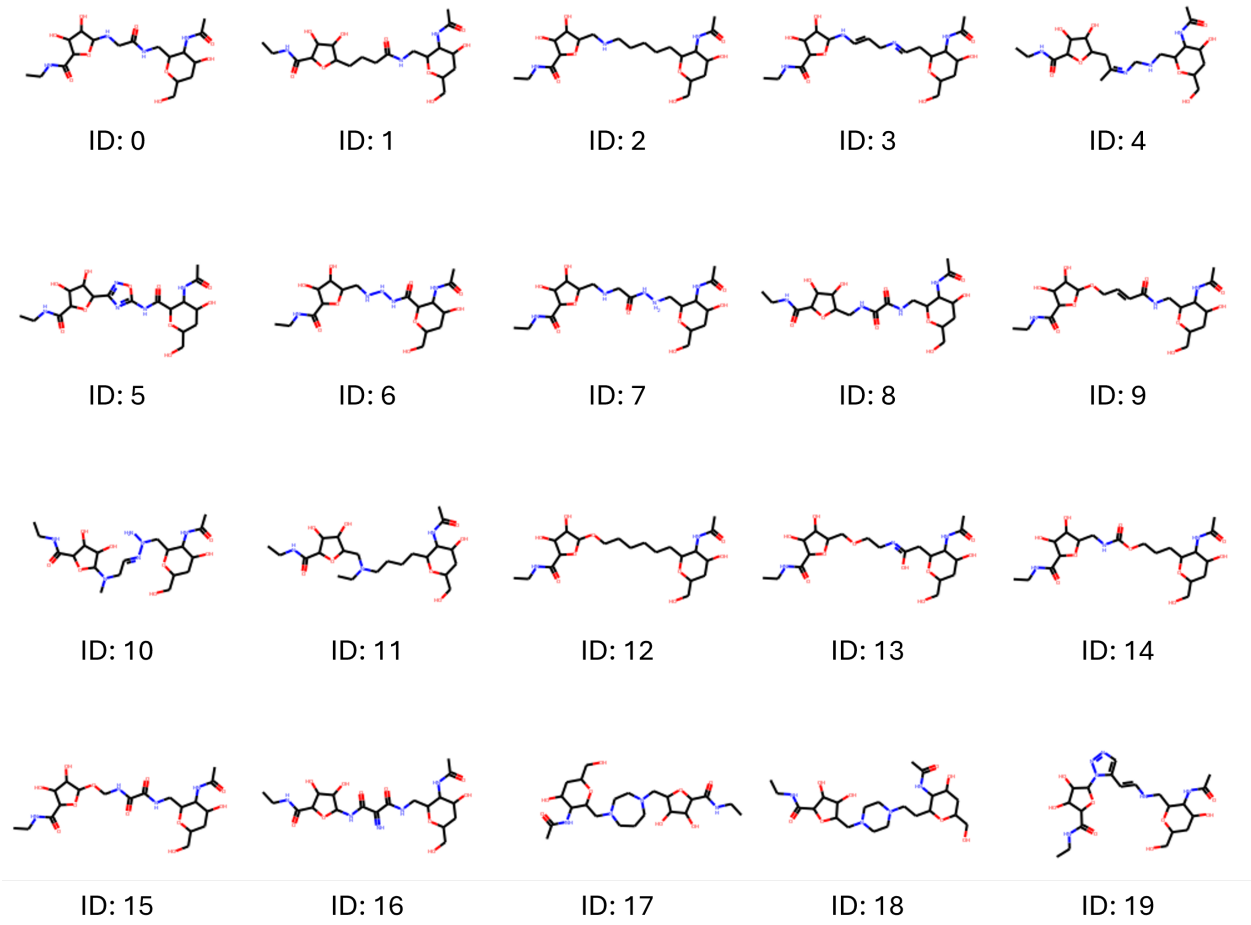


Figure 3.23: Molecular structures of 20 unique compounds with the number of heavy atoms in linker between 7 and 9.

This filtering resulted in 21 qualified candidates with favorable synthetic feasibility and screening properties. Among these, 20 compounds were unique by SMILES string (Figure 3.23).

3.5.3 Structural evaluation and selection

The 20 candidate molecules were evaluated based on two criteria:

- **SC Score**: a real number calculated using pharmacophoric feature similarity and the shape similarity between the generated molecule and the original ligand
- **Fragment RMSD**: root mean square deviation calculated between predicted and retained fragment conformations after alignment, after embedding 3D conformations

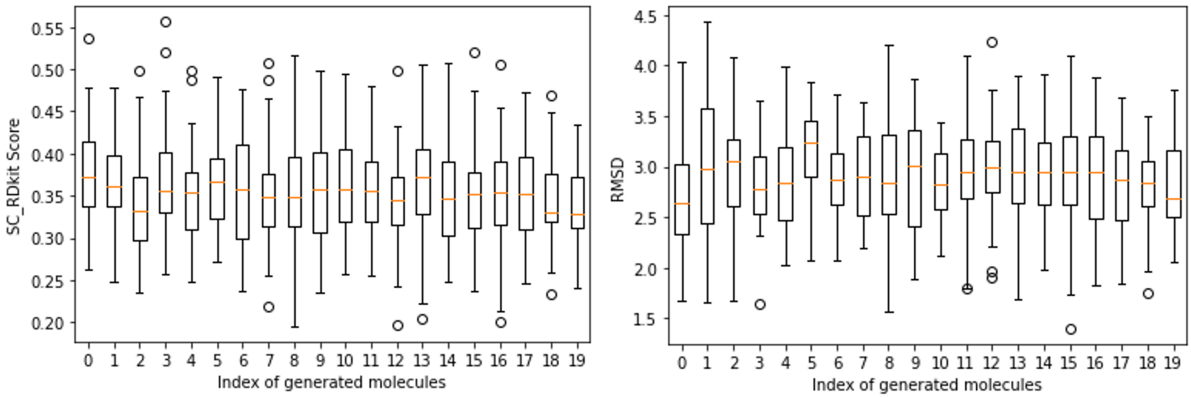


Figure 3.24: SC score and RMSD evaluation of 20 generated candidate compounds.

of each fragment.

Evaluation values are demonstrated in Figure 3.24, with no significant difference among these generated compounds. Generally, compounds with lower RMSD and higher SC score are preferred because they retain the spatial fidelity of the original fragment conformations (as indicated by low RMSD) while achieving favorable substructure connectivity and compactness (as measured by the SC score). This dual optimization ensures that the generated molecules not only preserve the geometry of predicted fragment placements but also form chemically coherent and synthetically plausible full ligands.

The selected compounds were docked into the M^{pro} binding pocket using AutoDock Vina. Among these, the compound shown in Figure 3.25 (Mol ID: 1, Pose 3) achieved a high predicted binding affinity of –6.40 kcal/mol as well as a desirable compound conformation where fragments are located at their assigned subpockets respectively in the initial setup.

3.5.4 Discussions

While fragment linking with generative models such as DeLinker offers a powerful means to construct full drug-like ligands from spatially compatible fragments, the overall process remains semi-manual and sensitive to human intervention. In this study, linker generation was applied to top-scoring fragment pairs at Center 2 and Center 3. However, this process can be repeated for other fragment combinations as more high-quality candidates are identified. Furthermore, the same process should be repeated for other center combinations.

It is important to emphasize that this design-evaluate-select loop where candidate fragments are chosen, docked, linked, filtered and re-evaluated, is not yet fully automated.

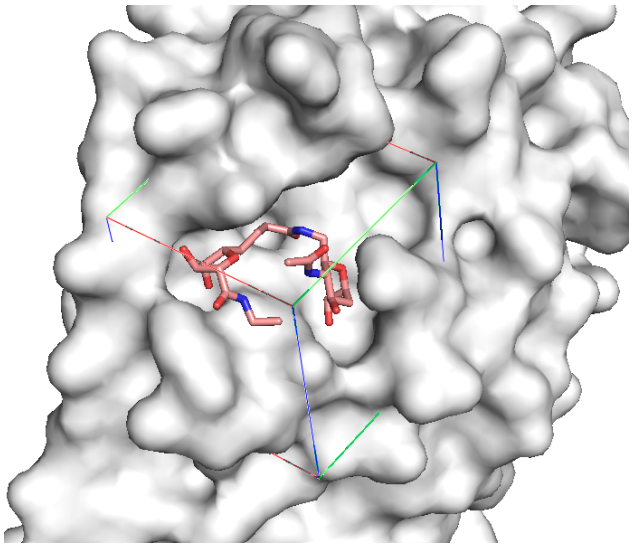


Figure 3.25: A hand picked generated compound whose docked pose aligns fragments to link with their initially assigned subpockets.

Each step requires domain intuition such as to define docking centers, interpret poses, select fragment pairs, and decide which linkers merit advancement. This manual oversight, while necessary under current limitations, introduces bottlenecks and potential biases that constrain scalability.

Moreover, the strategy inherently seeks local optima by linking one fragment pair at a time. Once a specific fragment pair is selected for linking, subsequent processes including fragment posing, atoms to link selection and linker generations explore a limited subspace around that chemical scaffold. For example, Figure 3.26 illustrate fragment docking does not always find the correct pose as in the full ligand due to the lack of geometry constraint by the linker, leading to an improper setup for DeLinker.

As a result, each intermediate result during this fragment linking pipeline may only reflect local solutions rather than globally optimals across the full chemical or conformational landscape. This underscores the broader challenge that the transition from promising fragment hits to well-optimized drug-like molecules is highly non-trivial. Future work will need to address both the automation and global search capabilities of fragment-to-ligand workflows to move closer to a truly end-to-end, data-driven drug discovery platform.

3.6 Conclusion and future perspective

This chapter presents an end-to-end exploration of ChemPLAN-Net, a deep learning framework designed to predict chemically relevant fragments that bind to local protein

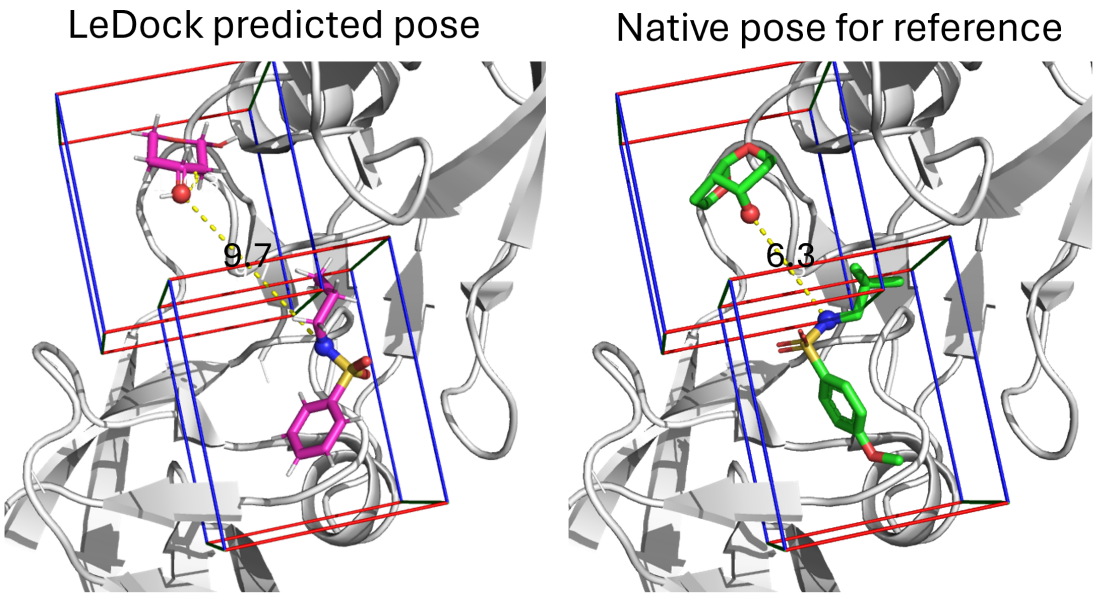


Figure 3.26: An example that fragment docking cannot predict the native orientation as in the full ligand, using HIV-1 protease with PDB ID: 6D0D.

environments. Building upon FragFEATURE and EnsFragFEATURE, ChemPLAN-Net reformulates fragment binding prediction as a binary classification task between local FEATURE descriptors and candidate ligand fragments. Its ability to predict binding fragments for unseen targets such as SARS-CoV-2 M^{pro}, CDK2, PTP1B, and selected molecular glue systems illustrates the potential of ChemPLAN-Net in FBDD. Beyond predictive modeling, this work demonstrates practical applications of fragment hits in drug discovery, via virtual compound library synthesis and screening, and structure-aware fragment linking using generative models such as DeLinker.

Despite these successes, the ChemPLAN-Net framework exhibits several limitations. Most notably, the model lacks explicit consideration of fragment orientation during training and inference. Because ChemPLAN-Net operates on unordered and vectorized descriptors without incorporating spatial placement or anchoring directionality for fragments. This omission hinders identification of full-ligand candidates for downstream drug design stages. As demonstrated in the virtual synthesis, high fragment atom coverage alone does not guarantee optimal docking or binding affinity.

While fragment-based docking and post-hoc structure-aware linker generation partially mitigate the orientation issue, they introduce a new set of challenges. The success of

fragment-to-ligand design becomes sensitive to hyperparameters such as initial fragment selection, conformer generation, linker length, and merging constraints. Without native integration of spatial reasoning into the fragment prediction stage, these downstream processes remain disjointed, limiting their effectiveness.

Looking forward, several improvements could be pursued to address these limitations. First, replacing fixed fingerprints with learnable and spatially grounded fragment encodings such as 3D graph neural networks could allow the model to capture orientation-aware environment-fragment compatibility. Second, incorporating protein–ligand complex structures from molecular dynamics simulations may enrich the training dataset with conformational ensembles that better reflect the binding landscape. Third, contrastive learning frameworks could help by further differentiating between ligand and environmental contributions, encouraging the network to learn context-aware fragment selection.

4 FeatureDock: A protein-ligand docking method guided by physicochemical feature-based local environment learning using transformer

This chapter presents FeatureDock framework, a novel and general feature-based deep learning pipeline for protein-ligand docking.

FeatureDock addresses several critical limitations in ChemPLAN-Net. First, ChemPLAN-Net exhibits limited generalization across diverse protein families, primarily because the relationship between local physicochemical environments and high-dimensional fragment fingerprints is highly complex and protein-family specific. Second, while ChemPLAN-Net effectively identifies binding fragments, it fails to resolve the fragment posing and fragment linking problem, as its predictions lack spatial orientation and remain confined to an intractably large fragment space.

To overcome these challenges, FeatureDock shifts from fragment-based learning to a pharmacophore-oriented representation, enabling robust generalization to novel protein targets without the need for protein-specific training or transfer learning. By learning local physicochemical environments using a transformer-based architecture, FeatureDock offers a scalable and interpretable approach for structure-based virtual screening. The model demonstrates high predictive performance in both ligand scoring and binding pose estimation, making it a promising candidate for integration into modern drug discovery workflows.

This chapter is adapted from a peer-reviewed publication "*Xue, M., Liu, B., Cao, S. et al. FeatureDock for protein-ligand docking guided by physicochemical feature-based local environment learning using transformer. npj Drug Discov. 2, 4 (2025). https://doi.org/10.1038/s44386-025-00005-6*".

4.1 Motivation from FBDD to pharmacophore-based drug discovery

As discussed in the Chapter 3, ChemPLAN-Net learns binding preferences between protein binding sites and ligand fragments from co-crystal structures, and successfully recovered known inhibitor fragments for proteases. ChemPLAN-Net circumvented the difficulties in multi-label multi-class classification by reducing the prediction task to binary classification

over environment–fragment pairs.

However, ChemPLAN-Net remains constrained by several key limitations. First, it lacks the ability to recombine predicted fragments into complete, drug-like molecules due to the absence of pose prediction. Furthermore, it cannot overcome the complexity in recombining fragments to an entire drug-like molecule from a chemical space spanned by 60k predefined fragments. This limits its ability to directly predict full ligands, generate poses, or score compounds. Second, the model must be trained separately for each protein family, owing to its poor generalization across diverse protein environments and fragment associations.

To overcome this, FeatureDock introduced two major changes. First, instead of predicting discrete fragments, it extracts more coarsely grained pharmacophore-based interactions from protein-ligand cocrystal structures. By formulating ligand binding in terms of coarse features (e.g. geometry preferences, hydrophobicity, hydrogen-bond donors/acceptors, etc.), FeatureDock avoids explicit enumeration over an unwieldy fragment library. This abstraction not only enables generalization beyond the constraints of any fixed fragment set but also facilitates the training of a unified model capable of capturing binding patterns across diverse protein families.

Second, FeatureDock encodes the local physicochemical environment around grid points on the protein surface instead of around functional centers. Specifically, FEATURE vectors are computed for regularly spaced grid points (at 1Å resolution) throughout the binding pocket. Each point is processed independently by a Transformer encoder, which predicts the probability of ligand atom occupancy or other pharmacophore-related properties. During training, the model learns to classify grid points as either "binding" or "non-binding" based on annotated co-crystal structures. The output forms a continuous probability density envelope over the pocket, which closely aligns with observed ligand atom distributions. This envelope serves as a guidance map for subsequent ligand pose prediction and scoring. This change accommodates FeatureDock for virtual screening.

In summary, the transition from a fragment-based to a pharmacophore-based learning paradigm in FeatureDock confers several advantages. It captures generalizable interaction patterns rather than enumerating specific fragment identities. Furthermore, moving from functional center encoding to grid-based representation enables the model to produce dense, spatially resolved interaction maps, supporting direct prediction of binding poses and compound scores.

4.2 Overview of docking methods and scoring functions

Structure-based molecular docking has long served as a foundational tool in computer-aided drug discovery (CADD), providing a cost-effective approach to predict the binding orientation and conformation of small-molecule ligands given a protein pocket. Traditional docking methods[249, 250, 251, 63], which typically follow a two-step paradigm involving pose generation followed by scoring, have evolved substantially over the past few decades.

4.2.1 Traditional docking approaches

The core of classical docking techniques is the scoring function, which aims to evaluate and rank the predicted ligand poses based on their estimated binding affinities. Early programs such as DOCK[252, 253, 254] and AutoDock[255] utilized physics-based scoring functions that incorporate terms like van der Waals interactions, electrostatics, and hydrogen bonding terms. These models attempt to capture the essential forces underlying molecular recognition. While they provide interpretable results grounded in physical principles, they are computationally expensive and often exhibit inaccurate treatment of polarization, solvation, and entropy effects[256, 257].

To enhance predictive efficiency and better reflect observed binding affinities, empirical scoring functions were introduced to address partial limitations in the physics-based docking methods by fitting parameters to the experimental data using regression[258, 259]. AutoDock Vina[260] represents a landmark in this space, using a combination of knowledge-based and empirical energy terms. Smina[261], a derivative of Vina, further enables users to define custom scoring terms, while other variants such as AutoDock VinaXB[262] and Vina-Carb[263] adapt Vina to specific chemical interactions such as halogen bonding and carbohydrate recognition.

These docking methods typically employ genetic algorithms (GA)[252, 253, 255] or Monte Carlo (MC) annealing[260, 262, 263, 264, 265, 266], often in conjunction with gradient-based optimization techniques, to efficiently sample and refine ligand binding poses within the protein pocket.

Despite these advancements, traditional scoring functions remain prone to high false positive rates during virtual screening. They frequently fail to distinguish strong binders from weak ones in large chemical libraries, undermining their utility in early-phase drug discovery[267]. Benchmark studies like CASF-2016[268] reveal relatively low Pearson correlation coefficients (R^c) between docking scores and experimental binding affinities for many widely-used programs, including AutoDock Vina ($R^c = 0.604$), GOLD[269] ($R^c = 0.416$–0.617), MOE[270] ($R^c = 0.405$–0.591), and Glide[271, 272] ($R^c = 0.467$–0.513), where

the ranges indicate more than one scoring function exists in the software. These results illustrate the intrinsic limitations of conventional scoring functions in reliably prioritizing true binders.

4.2.2 Machine learning-based scoring functions

To overcome the poor predictive power of traditional scoring functions, machine learning approaches have emerged as powerful alternatives[249, 268, 273]. These models aim to learn complex, nonlinear mappings between structural features of protein-ligand complexes and binding affinities. For example, RF-Score[274] uses random forests and hand-crafted atom-pair descriptors. KDEEP[170] and ACNN[187] employs 3D convolutional neural networks on voxelized protein-ligand complexes. PotentialNet[275], a distance-aware graph neural network, further captures topological and spatial relationships in molecular graphs.

While these methods significantly improve the correlation between predicted and experimental affinities, especially on curated datasets like PDBBind dataset[276], they are predominantly designed to re-score existing docked poses. That is, they rely on an accurate 3D configuration of the protein-ligand complex as input, rather than propose poses themselves.

4.2.3 Deep learning-based pose prediction models

Recent advancements have led to deep learning models capable of directly predicting ligand binding poses without explicit pose sampling or scoring. EquiBind[277] and TankBind[278] adopt $E(3)$-equivariant graph neural networks to effectively represent both the protein and ligand structures. These models predict rigid-body $SE(3)$ transformations to align ligands into the protein binding pocket, achieving pose predictions in a single inference step. TankBind additionally incorporates trigonometric constraints to enhance geometric fidelity, and both methods include torsion angle refinements for flexible docking.

An alternative approach is exemplified by DiffDock[226], which frames docking as a generative task using denoising diffusion probabilistic models. Starting from randomized ligand structures, DiffDock iteratively refines them into bound poses. Moreover, it introduces a confidence scoring mechanism to assess the quality of predicted poses.

Despite their impressive efficiency and generalizability, these deep learning methods focus primarily on pose prediction rather than binding affinity estimation. Consequently, while they often outperform traditional docking in RMSD-based pose recovery, they lack

robust scoring functions to effectively distinguish strong binders from weak ones in virtual screening.

4.2.4 Motivation of FeatureDock from a docking perspective

To address the gap between the high false positive limitations in traditional docking methods due to weak scoring functions and the absence of posing ability in most deep learning scoring functions, we introduce FeatureDock to score and pose compounds in one physicochemical feature-based framework.

We outline the pipeline for our FeatureDock framework in Figure 4.1. The potential ligand-binding spaces collected from in the PDBBind v2020 refined set[276] are firstly discretized into grid points, embedded using 3D-invariant FEATURE[235] representations, which capture the physicochemical and geometric properties of the local protein environment. These representations have been shown sufficiently descriptive to extract the local protein information, as demonstrated in Chapter 3. Then, the state-of-the-art Transformer encoder[202] is adopted to predict the binding probabilities of grid points in the query pockets, achieving higher accuracy than the models of feed forward neural network (FNN) and Convolutional neural network (CNN) of the same capacity. Consequently, FeatureDock outputs probability density envelopes formed by grid points in the query space with their predicted probabilities.

Importantly, due to exploiting the similarity of local protein information, FeatureDock shows robust generalization ability on predicting non-homogeneous novel protein structures that are not included in the model training. Furthermore, it demonstrates that our model can predict ligand-explorable regions with higher probabilities. Last but not least, the attention mechanism in Transformer further helps explain and visualize the importance of input features by extracting the attention weights.

With the probability density envelopes, we design a novel scoring function combined with a position optimization algorithm to guide compound posing and scoring. Our custom scoring function, incorporating the predicted probability density envelopes and compound coordinates, shows its prominent performance in virtual screening. Additionally, FeatureDock enables to predict compound poses via optimization, extending its capabilities beyond most machine-learning based scoring functions. We will describe the dataset curation, model training and virtual screening in detail in Section 4.3.

After training full models using all available protein structures, we validate FeatureDock's capabilities to select strong inhibitors from ChEMBL bioassay datasets[279] for two different protein systems: inactivated Cyclin-Dependent Kinase 2 (CDK2) and Angiotensin-

converting enzyme (ACE). The scoring function of our method, derived from the predicted probability density envelope, exhibits a superior ability to differentiate between strong and weak inhibitors compared to DiffDock[226], Smina[261] and AutoDock Vina[260], as evaluated through the KL divergence and AUC evaluation. Notably, our model can accurately retrieve the binding poses of top-predicted ligands through the L-BFGS-B optimization process based on our scoring function. These binding poses closely approximate their native structures in cocrystal configurations, demonstrating a high level of proximity (RMSD $\approx$ 2Å).

4.3 FeatureDock pipeline

4.3.1 Dataset curation

FeatureDock curated the training set from the PDBBind v2020 refined dataset, which provides over 5,300 high-quality, non-covalently bonded protein–ligand cocrystal structures. For each complex, ligand-binding pocket is defined by taking the convex hull of protein residues within 6Å of the ligand. This pocket volume is discretized into a 3D grid of points spaced 1Å apart. To focus more on relevant regions, grid points closer than 1Å or farther than 6Å from any protein atom are discarded since ligand atoms are highly unlikely in these extreme regions.

The choice of distance parameters to determine binding pockets, grid resolution and grid point filtering are subjected to the training data and computational resources. For example, when applying the same pipeline to pockets that differ dramatically from small-molecule clefts, e.g. compounds binding at PPIs, the choice of pocket regions, the width of shells, and the span of local environments can be enlarged to adapt for the flatter and broader interface.

4.3.1.1 Protein representation and data labeling

Each grid point is featurized using FEATURE vectors as described in Table 3.3, which encode a comprehensive set of 80 physicochemical properties (atomic types, residue types, secondary structure, etc.) across 6 concentric spherical shells, with a width of 1.25Å capturing up to 7.5Å radius. As a result, each grid point is represented by a tensor of shape 6×80 (480 features in total).

Since these features are invariant to rotations and translations, no additional alignment or augmentation is required. Figure 4.2 demonstrates that model predictions remain consistent under a 1.5Å grid shift along the z-axis, confirming that the learned probability

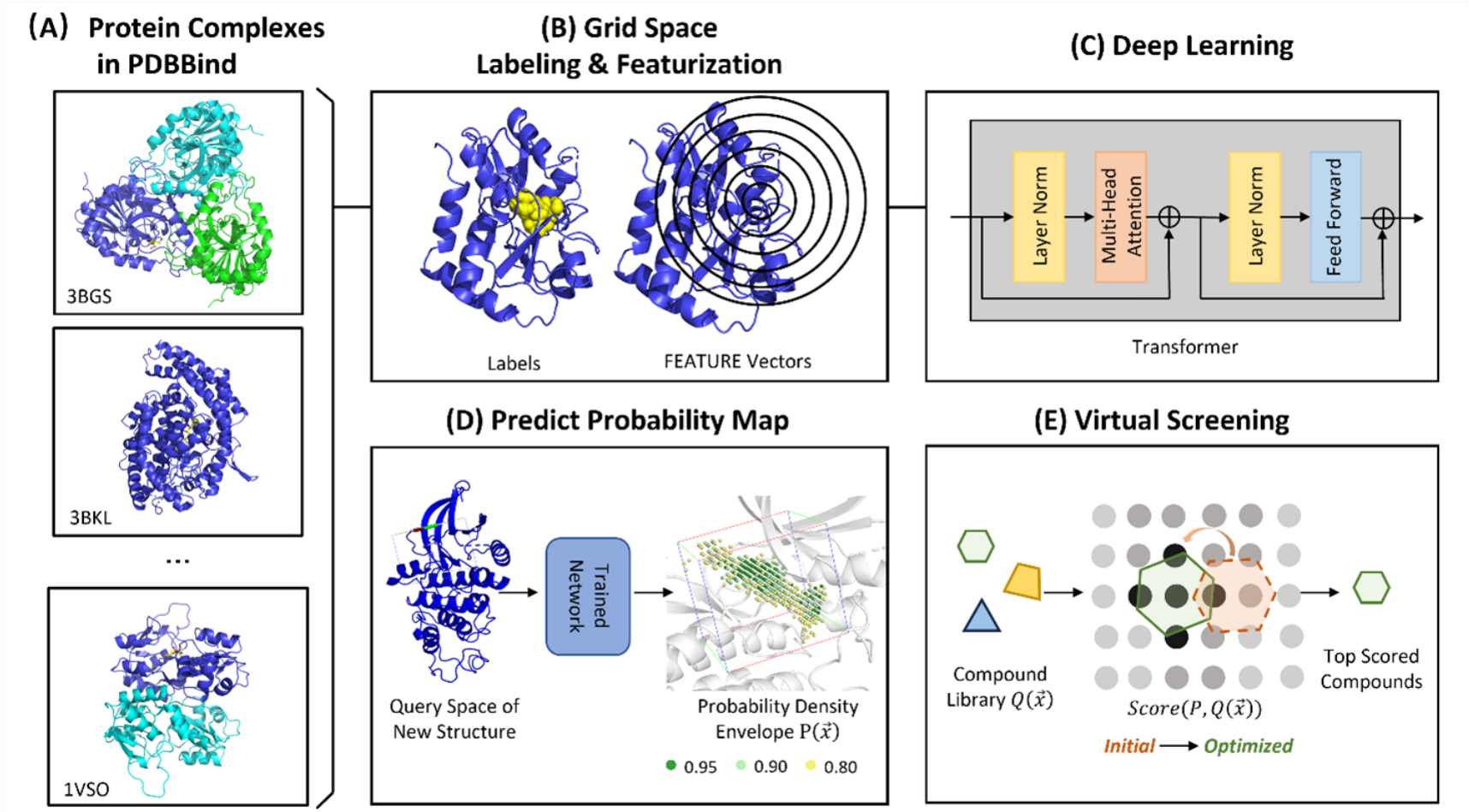


Figure 4.1: FeatureDock pipeline. (A) Collect protein-ligand complexes from PDBBind v2020 refined set. (B) Extract and featurize protein local environments around grid points in the ligand-binding pocket and then label the grid points as either binding or non-binding with the ligand. (C) Train the Transformer Encoder to predict the ligand-binding probability of each grid point. (D) Predict the probability density envelope for the query space in apo proteins using the trained model. Grid points with darker color are more likely to be occupied by compounds. (E) Apply the predicted probability density envelope to virtual screening and pose prediction.

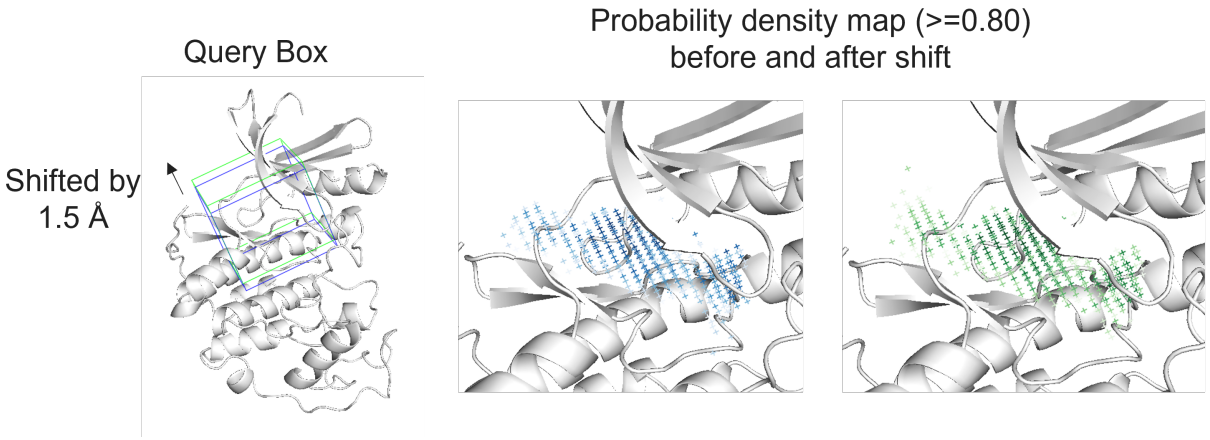


Figure 4.2: Sensitivity test of FeatureDock's probability density map to the selection of grid point discretization. (A) Green query box of 1B38 (inactive CDK2) is shifted along z-axis from the original blue query box. (B) The predicted probability envelopes using the original query box (left) and the shifted query box (right).

field is a property of local environments, not absolute coordinates. Especially, the two resulting probability density envelopes are visually and quantitatively indistinguishable, except that the 0.5Å non-overlap shift caused by the discretization resolution rather than the 1.5Å replacement of the box.

Each grid point is labeled according to the presence of ligand atoms. Grid points are assigned by a positive (binding) label if any ligand heavy atom lies within 1.5Å, and negative (non-binding) otherwise. This binary label captures whether the site is likely to be occupied by the ligand in the crystal structure. The labeling scheme can be flexibly extended to capture other interactions. For example, the program supports labelling grid points by the presence of pharmacophoric group (e.g. hydrogen-bond donor/acceptor) or other interaction criteria (e.g. electrostatic properties)[117]. By choosing different labeling definitions or multiple labels per point, the same dataset and model framework can support multi-task learning setups.

4.3.1.2 Dataset split

Complexes were partitioned by protein sequence identity rather than individual structures. All PDBBind entries are clustered with MMSeq2 at a 90% identity threshold, yielding approximately 1300 non-overlapping clusters (Figure4.3, left). During training, 10% of clusters were withheld as a validation set, ensuring no homologous proteins or their ligands are shared between training and validation. Beyond the basic split, we further test

generalization by excluding all complexes of a given protein (e.g. all Cyclin-Dependent Kinase 2 structures) from the training clusters at the stage of hypothesis validation to mimic the scenario of applying the trained model to predict novel protein structures. In this case, FeatureDock generalizes across various proteins, compared to ChemPLAN-Net which requires to train individual models for a specific protein family. This strategy enforces strict generalization, particularly when testing on unseen protein families.

Importantly, this cluster-based split also minimizes chemical space overlap among ligands during training. Because similar proteins often bind similar ligands, segregating by protein clusters helps ensure that chemically similar molecules are likewise separated, thus avoiding data leakage from ligand. In other words, the selected validation clusters exclude not only the homologous proteins but also their associated ligands. As a result, closely-related protein–ligand binding space are absent during training as a whole. Figure 4.4 illustrates this effect by the T-SNE reduced space of ligands. It demonstrates that ligands within the same protein sequence cluster exhibit contiguous T-SNE embedding, validating the effectiveness of the clustering-based split.

In summary, this sequence-identity clustering and family leave-out protocols rigorously reduce potential data leakage and require the FeatureDock model to predict for novel protein-ligand interactions, thus providing a conservative and robust estimate of its true predictive power.

4.3.1.3 Dataset imbalance and resampling

The training dataset exhibits two prominent imbalances:

- **Protein cluster imbalance**. Some homologous protein clusters (e.g., kinase proteins) possess dozens of cocrystal structures, whereas most targets are represented by a single entry. (Fig.4.3).
- **Label imbalance**. Within every pocket, non-binding voxels overwhelm binding voxels by roughly an order of magnitude.

If left unchecked, these skews would encourage the network either to predict the majority class (negative class) everywhere or to over-specialize on the most common protein clusters. Therefore, FeatureDock adopts a two-level resampling scheme executed on-the-fly for every mini-batch during training:

- **Cluster-level balancing**. Proteins are first grouped into sequence clusters by 90% identity. During batching, clusters instead of individual structures are sampled uni-

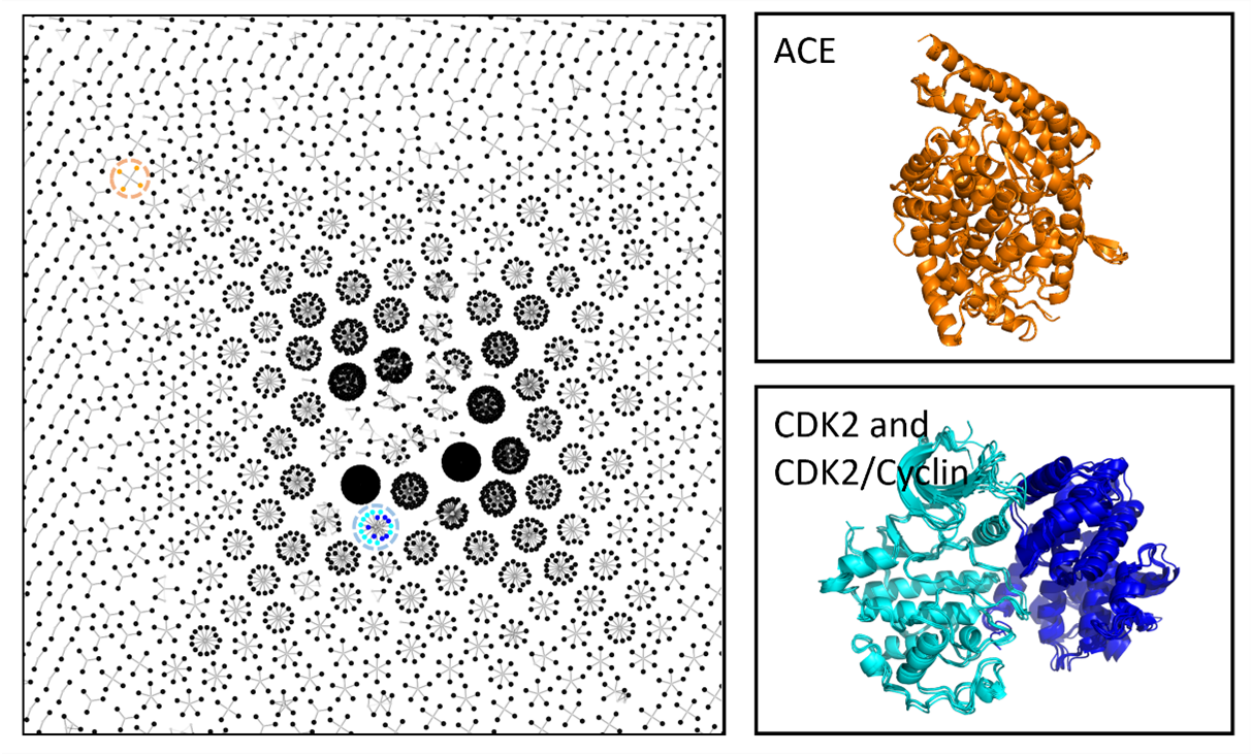


Figure 4.3: Structure clustering based on MMSeq2 which is used to split dataset during model training. Left Panel: Protein clustering based on 90% sequence identity. Right panel: two examples of the clusters. ACE structures are clustered and colored in orange. Inactive CDK2 structures (colored in cyan), which is connected to the active CDK2 with Cyclins (colored in dark blue).

formly. This guarantees that local environments from rare protein families contribute as often as the abundant ones.

- **Label-level balancing**. From each selected structure, a fixed number of grid points are sampled to form a 1:1 ratio of positives to negatives. Scarce positives are oversampled (with replacement), whereas the excess negatives are undersampled.

In summary, by adopting this dynamic resampling strategy, the model generalized across diverse protein structures by learning from less abundant protein clusers. Moreover, the model learns a meaningful binding propensity score rather than a trivial negative predictor by explicitly oversampling minority classes and under-sampling majority classes.

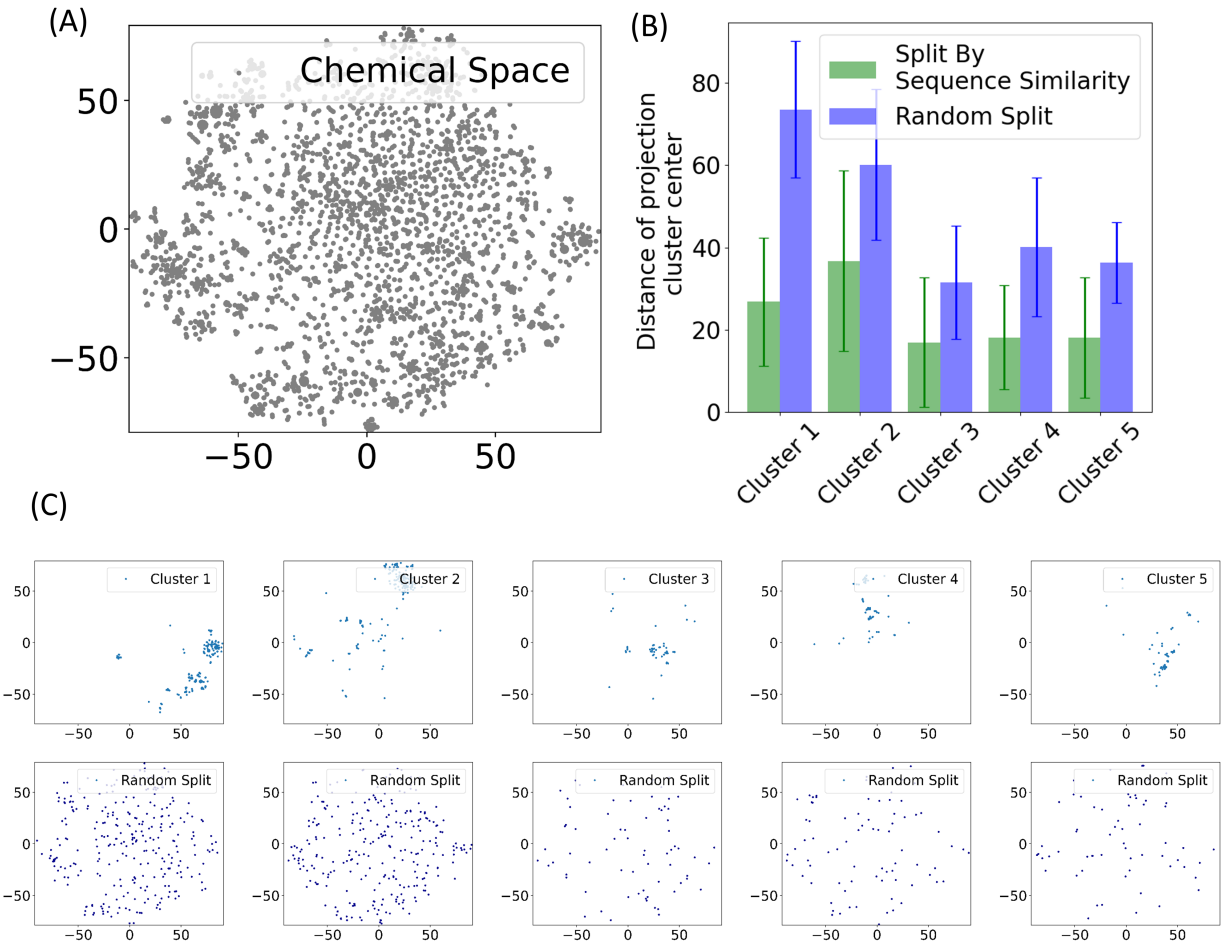


Figure 4.4: Ligand similarity validation in the dataset split during model training. (A) T-SNE projected 2D chemical space. (B) Cluster radius of five largest clusters obtained from stratified split and compared to the random split. The cluster radius is defined as the average distance between ligands to the cluster center on the T-SNE projected 2D space. (C) Span by ligands in five largest clusters by protein sequence similarity, colored by different sequence clusters vs span by randomly selected ligands of the same number.

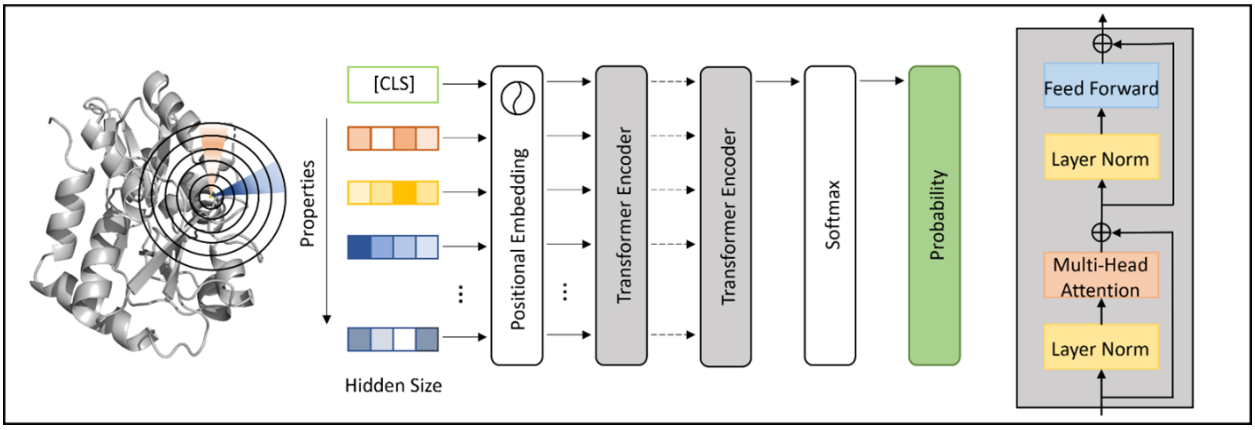


Figure 4.5: Transformer encoder architecture.

4.3.2 Neural network architectures

The core of FeatureDock is a deep neural network that maps each grid point to a probability indicating ligand preference. We compared several architectures, including fully-connected (Feed Forward) networks (FNN), convolutional neural networks with residual connections (ResNet), against a Transformer Encoder which we chose as the final model. In all cases, the input is a 6×80 tensor flattened or reshaped, and the output is a single probability per grid point. Each architecture together with the information it explicitly captures is described as follows.

4.3.2.1 Transformer encoder

Each grid point is encoded as 80 features over six shells. The Transformer encoder model treats each physicochemical property as a word in a sentence. As a result, each grid point contains 80 words, in addition to a prepended special token "[CLS]" used for the sequence-level classification (Figure 4.5). Each sequence is processed by a stack of Transformer encoder blocks. As discussed in Section 2.2.5, each encoder block contains a multi-head attention (MHA) block, followed by a feed forward neural network, with residual connections around each. The Transformer encoder uses the all-to-all self-attention mechanism to compute weighted combinations of all features to the classification token "[CLS]", awared of the rich context of local environment. The attention scores from each physicochemical property to the "[CLS]" token are post-analyzed to explain the feature attributions.

4.3.2.2 Benchmark architectures

- **Feed forward neural network (Figure 4.6A)**. The 6×80 input tensor is first flattened into a 480-dimensional vector. This vector is batch-normalized to standardize the feature distribution, then fed into a series of fully connected (dense) layers. Each dense layer computes a linear combination of its inputs followed by a nonlinear activation (e.g. ReLU), producing a new hidden feature vector $h_{i+1} = \mathrm{ReLU}(W_i h_i + b_i)$. After a stack of hidden layers, a final linear layer projects to the output space, and a softmax function converts the result to a probability. In this architecture, each output unit depends on the entire input vector through learned weight matrices, but there is no explicit modeling of spatial or sequential context among the 480 inputs beyond what the network can learn implicitly.

- **ResNet (Figure 4.6B)**. The 6×80 input is reshaped into a $80 \times 6 \times 1$ tensor (viewed as a 80-channel "image" of height=6 and width=1). This tensor passes through a series of convolutional blocks. In each block, a convolution with kernel size 2×1 that capture local patterns along the six-shell dimension and 80 output channels, followed by batch normalization and a non-linear activation. Crucially, the original block input is added to the output via a residual connection. After the convolutional blocks, the resulting tensor is flattened and pooled, fed into a softmax layer for classification. The ResNet thus explicitly exploits local spatial pattern across adjcent shells and leverages residual links to support deeper architectures.

4.3.3 Evaluation metrics

Multiple metrics are used to capture different aspects of performance when evaluating the prediction performance of FeatureDock and two benchmark models, because single measure may not fully captures model quality in classification tasks, especially when class frequencies are skewed. For example, classification accuracy alone can be misleading on imbalanced data as a trivial classifier that always predicts the majority class may obtain high accuracy despite failing to detect any minority-class instances.

To address this, multiple metrics are adopted to achieve more robust evaluation, including cross-entropy loss, F1 score (combining precision and recall), MCC (Mathews correlation coefficient) and AUROC (Area Under the Receiver Operating Characteristic curve), in addition to accuracy, precision and recall. The metrics of accuracy, precision, recall, F1 score and MCC all derived from the confusion matrix (Table 4.1).

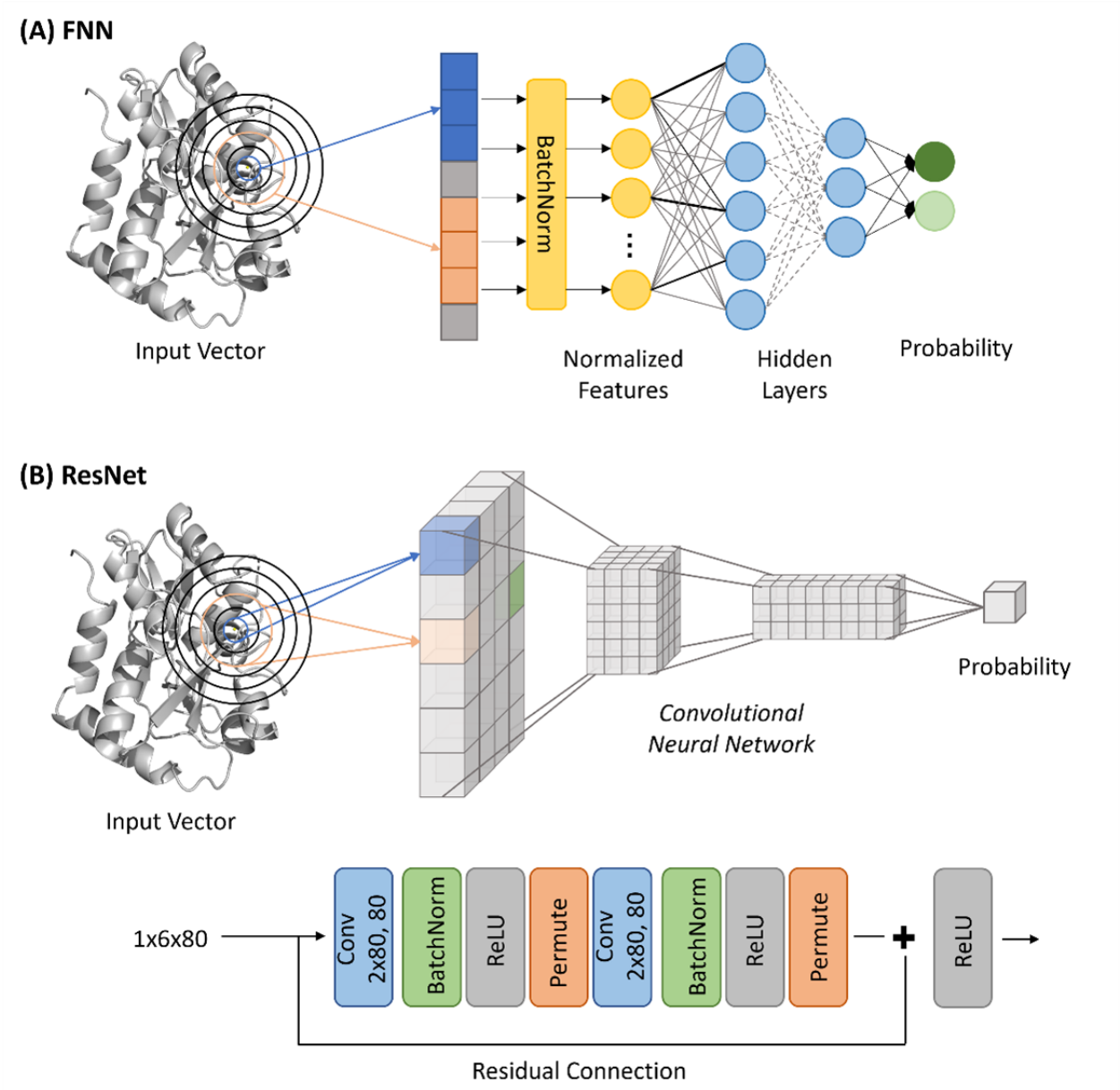


Figure 4.6: Benchmark architectures. (A) FNN takes the flattened vector (480-dimension) as input, followed by a BatchNorm layer and a series of fully-connected layers. (B) ResNet takes the 1x6x80 tensor as input, followed by a series of ResNet blocks

Table 4.1: Confusion matrix

		Predicted Labels	
		0	1
True Labels	0	TN	FP
	1	FN	TP

Each evaluation metric is defined as follows:

- **Cross entropy loss**. For a model with C mutually exclusive classes, the data point i is labeled as $\mathbf{y}_i = (y_{i,0}, y_{i,1}, \ldots, y_{i,C-1})$ where $y_{i,k} = 1$ and $y_{i,j\neq k} = 0$ if this data point belongs to the class k. The model output a probability vector $\mathbf{p}_i = (p_{i,0}, p_{i,1}, \ldots, p_{i,C-1})$ normalized to 1 after Softmax activation. The categorical cross entropy loss is defined as

$$\mathcal{L} = -\sum_{i}^{N}\sum_{j=0}^{C-1} y_{i,j} \log p_{i,j} \tag{4.1}$$

  Especially, when $C = 2$, this general formula reduces to the binary cross entropy loss:

$$\mathcal{L} = -\sum_{i}^{N} [y_i \log p_i + (1 - y_i) \log(1 - p_i)] \tag{4.2}$$

  The cross entropy loss is a consistent metric as the maximum loglikelihood estimation.

- **Accuracy, Precision, Recall**. Accuracy is defined as the ratio of true positives (TP) and true negatives (TN) among all samples:

$$\mathrm{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{4.3}$$

  When classes are imbalanced, it is important to separately consider precision (the ratio of actual positives versus positive predictions) and recall (the ratio of positive predictions versus the actual positives):

$$\begin{aligned} \mathrm{Precision} &= \frac{TP}{TP + FP} \\ \mathrm{Recall} &= \frac{TP}{TP + FN} \end{aligned} \tag{4.4}$$

  In practice there is often a trade-off between precision and recall when choosing the proper predicted probability cutoff value because raising the decision threshold will

increase precision (fewer false positives) but lower recall (more false negatives), and vice versa.

- **F1 score**. To combine precision and recall into a single metric, F1 score is defined as the harmonic mean of precision and recall:

$$\text{F1 score} = \frac{2}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{4.5}$$

  F1 score better reflects the predictive power of models over accuracy on class-imbalanced dataset.

- **MCC**. The Matthews correlation coefficient (MCC) is a more balanced and comprehensive metric[280] that uses all four entries of the binary confusion matrix, focusing not only on the positive class but also the negative class:

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \tag{4.6}$$

  MCC ranges between -1 and $+1$, where $+1$ indicates perfect prediction, 0 indicates no better than random, and –1 indicates completely wrong.

- **AUROC**. Unlike confusion matrix and metrics derived from the confusion matrix which is evaluated under a certain cutoff, the ROC shows the trade-off between detecting positive examples and avoiding false alarms over all thresholds. The area under the ROC curve (AUC) summarizes this plot (Figure 4.7) as a value $\in [0, 1]$. A perfect classifier has an AUC of 1.0, whereas a random classifier yields an AUC of 0.5. Classifiers with $\text{AUC} < 0.5$ perform worse than the random classifier.

4.3.4 Hyperparameters

For all three model architectures, the training starts with a learning rate of 0.01 using AdamW[281] optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ by minimizing the cross entropy loss. The learning rate decays with a factor of 0.5 whenever the validation loss has been plateaued for 2 epochs, down to a minimum learning rate of 10^{-6}.

During model size and architecture selection, no weight decay or other regularization terms are applied to reflect the raw capacity of each architecture. The default training epochs was set to 30 in the leave-out models reported in the result Sections 4.4.1 and 4.4.5, and 50 in the full model which is used in the application Sections 4.5 and 4.6. The best

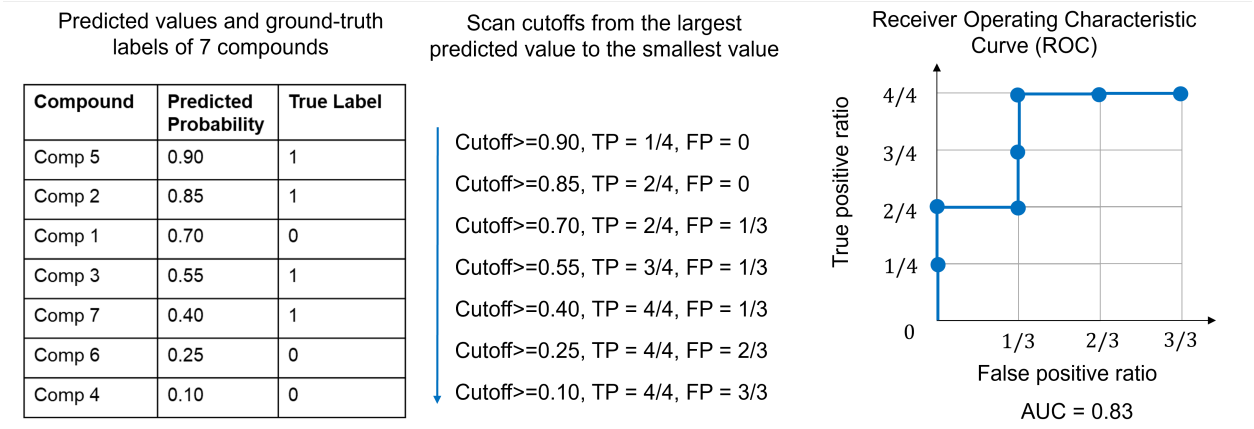


Figure 4.7: A toy example to explain AUROC

model parameters are recorded based on the validation loss to avoid overfitting. In practice, models converged well within these epoch limits, and further training showed negligible improvements.

Each model size and architecture combination was trained on five random training/validation splits based on data split methods described in Section 4.3.1.2, providing robust validation statistics for capacity selection. Each mini-batch was comprised 5,000 grid points, obtained by sampling five protein structures per protein cluster and 1,000 class-balanced data points per structure as described in Section 4.3.1.3. All experiments were executed on NVIDIA A100-SXM4 GPUs equipped with 80 GB of memory.

4.4 Results

4.4.1 Model training and selection

For each architecture, we conducted an extensive grid search (Figure 4.8) over four model capacities, five initial learning rates, whether or not to apply the plateau-based learning-rate decay. In total, 40 configurations ($4 \times 5 \times 2$) were evaluated per architecture (FNN, ResNet, Transformer), resulting in 120 models overall. Among these, the Transformer encoder comprising 500k parameters and an initial learning rate of 0.01 with plateau learning rate decay achieved the lowest cross entropy loss on the validation set.

Fixing this learning rate schedule, we further compared different architectures across multiple metrics discussed in Section 4.3.3. Not suprisingly, the models with residual connections (ResNet and Transformer) exhibited more stable performance across model

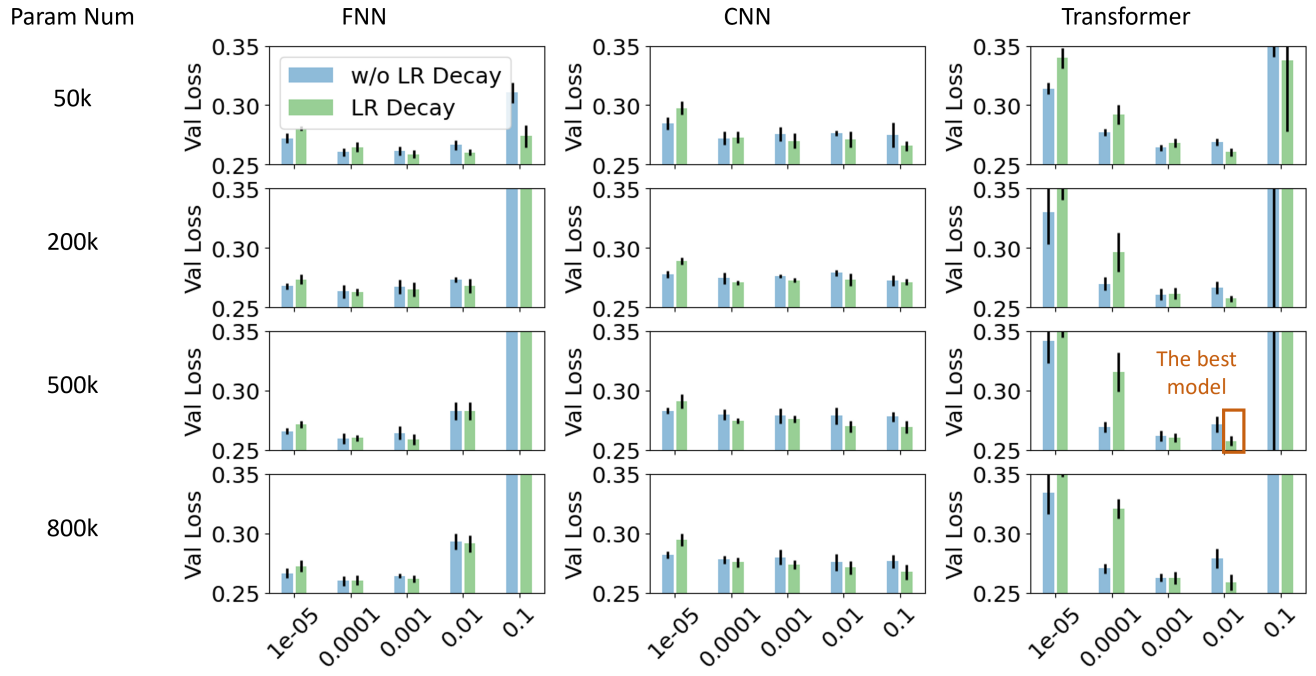


Figure 4.8: Comprehensive hyperparameter tuning results. For each model architecture and model size described in Table S3, the hyperparameter of learning rate is tuned from 0.00001 to 0.1 either with or without a learning rate decay. A Transformer model (circled by orange) with 20 blocks/500k parameters, initial learning rate of 0.01 with plateau learning decay, delivers the best performance.

sizes. Moreover the Transformer not only achieved the lowest average validation loss but also demonstrated lower variance across different runs (Figure 4.13A), establishing it as the default engine within FeatureDock. Specifically, the Transformer encoder with ~500k is selected because it balances the model size and performance on the curated dataset. Detailed values are listed in Tables 4.2. The values of the best configuration are highlighted in bold.

4.4.2 Predictive power across diverse functional groups of ligands and protein clusters

The model performance is further decomposed to various functional groups of ligands, in order to demonstrate the generality of the Transformer model across diverse ligand chemotypes. Ten representative functional groups that collectively cover 82.3% of ligand heavy atoms were analyzed individually (Figure 4.9). Performance remained stable across all functional groups, indicating that the model achieves predictive power over a broad chemical space.

Moreover, we also examined whether family-specific fine-tuning offers any performance advantage. Using a leave-out strategy for CDK2 and its homologs, we fine-tuned the general model on other kinase structures. Results (Table 4.3) revealed negligible gains from fine-tuning, suggesting that the general model already captures broad, transferable binding features without requiring family-specific adjustments.

4.4.3 Predictive power compared to other docking methods

To further assess the predictive power of FeatureDock against other docking methods, including AutoDock Vina, AutoDock4, and DiffDock, we evaluated the performance on the held-out validation set (approximately 660 protein–ligand complexes) of one trained model. For each ligand, FeatureDock generated a probability-density envelope, whereas the docking method to compare furnished its ten highest-ranked docking poses. These poses were converted to grid-based occupancy maps so that all approaches can be assessed on using metrics defined on probabilities.

Model performance was quantified in two ways. First, binary cross-entropy between each predicted probabilities and the ground-truth ligand occupancy was calculated. Lower cross-entropy values indicate closer agreement. As depicted in Figure 4.10A, FeatureDock attained the smallest cross-entropy, surpassing all baselines and underscoring its superior pose localization. Second, F1 scores across probability cutoffs ranging from 0.5 to 0.9 were calculated. DiffDock yielded the highest F1 score overall, yet FeatureDock consistently

Table 4.2: Model performance on the validation set after class-balancing across different sizes and architectures (mean ± std over five random runs).

Model Size	Architecture	Loss	AUC	Precision	Recall	F1 score	Accuracy	MCC
~50k	FNN	0.278 ± 0.004	0.880 ± 0.002	0.845 ± 0.003	0.930 ± 0.006	0.886 ± 0.003	0.880 ± 0.002	0.764 ± 0.005
	ResNet	0.276 ± 0.006	0.881 ± 0.003	0.852 ± 0.007	0.922 ± 0.015	0.885 ± 0.004	0.881 ± 0.003	0.764 ± 0.007
	Transformer	0.272 ± 0.009	0.882 ± 0.004	0.841 ± 0.004	0.942 ± 0.009	0.889 ± 0.004	0.882 ± 0.004	0.770 ± 0.009
~200k	FNN	0.283 ± 0.015	0.879 ± 0.009	0.848 ± 0.012	0.925 ± 0.025	0.884 ± 0.010	0.879 ± 0.009	0.762 ± 0.019
	ResNet	0.287 ± 0.004	0.876 ± 0.004	0.848 ± 0.007	0.916 ± 0.020	0.881 ± 0.006	0.876 ± 0.004	0.754 ± 0.009
	Transformer	0.268 ± 0.002	0.884 ± 0.002	0.852 ± 0.003	0.931 ± 0.005	0.890 ± 0.002	0.884 ± 0.002	0.772 ± 0.004
~500k	FNN	0.289 ± 0.014	0.880 ± 0.005	0.830 ± 0.015	0.955 ± 0.015	0.888 ± 0.004	0.880 ± 0.005	0.769 ± 0.007
	ResNet	0.284 ± 0.005	0.878 ± 0.004	0.848 ± 0.014	0.923 ± 0.023	0.884 ± 0.005	0.878 ± 0.004	0.761 ± 0.008
	Transformer	$\mathbf{0.269 \pm 0.004}$	$\mathbf{0.885 \pm 0.002}$	$\mathbf{0.848 \pm 0.004}$	$\mathbf{0.937 \pm 0.009}$	$\mathbf{0.890 \pm 0.002}$	$\mathbf{0.885 \pm 0.002}$	$\mathbf{0.773 \pm 0.005}$
~800k	FNN	0.295 ± 0.006	0.874 ± 0.001	0.819 ± 0.007	0.962 ± 0.013	0.884 ± 0.002	0.874 ± 0.001	0.761 ± 0.004
	ResNet	0.286 ± 0.010	0.878 ± 0.005	0.852 ± 0.016	0.916 ± 0.019	0.882 ± 0.004	0.878 ± 0.005	0.759 ± 0.008
	Transformer	0.277 ± 0.006	0.881 ± 0.004	0.845 ± 0.005	0.932 ± 0.007	0.886 ± 0.004	0.881 ± 0.004	0.765 ± 0.008

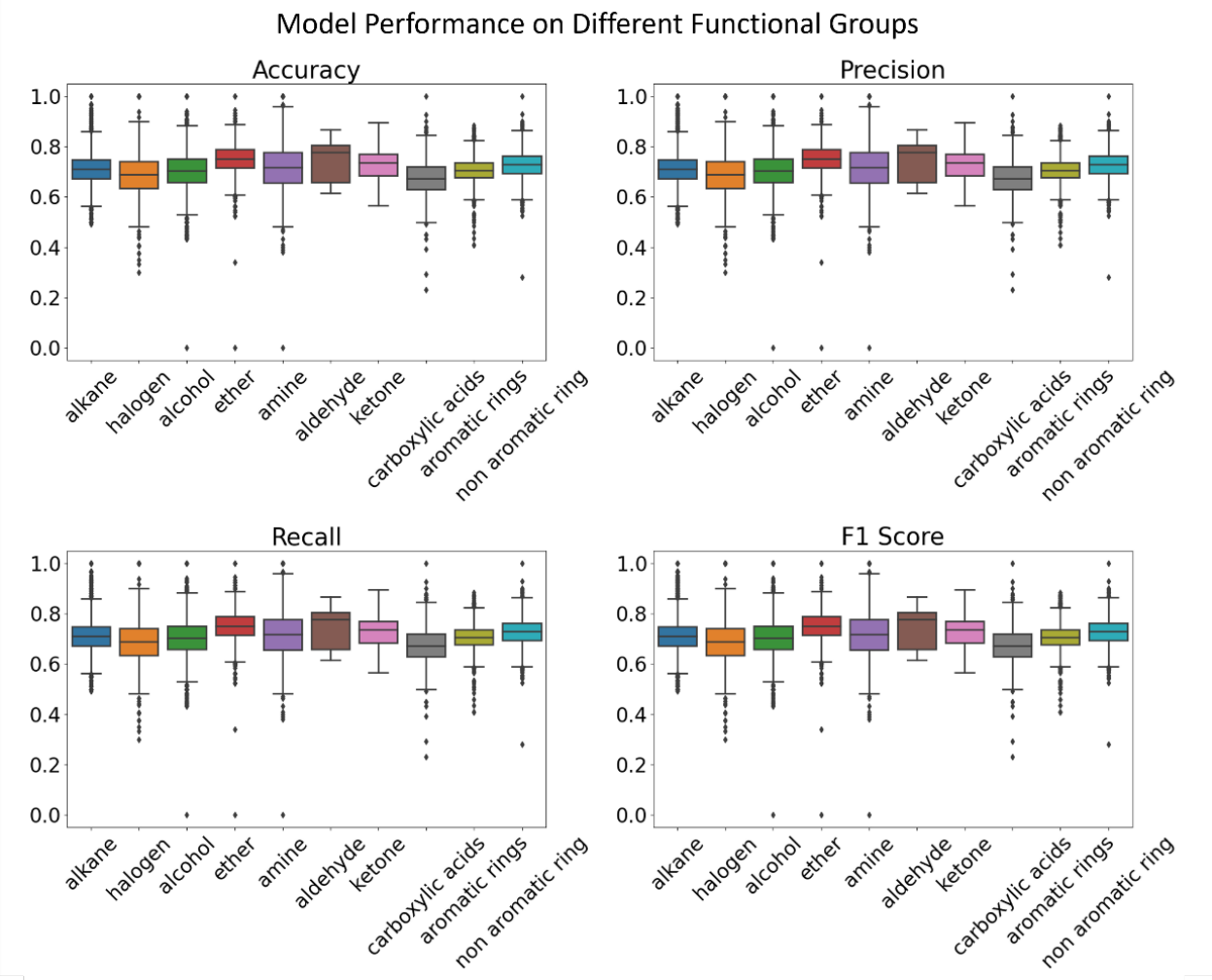


Figure 4.9: Model performance across functional groups.

Table 4.3: Fine-tuning results on kinase dataset (CDK2 leave-out model) using 500k-parameter (20-block) Transformer encoder model.

	~500k Transformer						
	Loss	**AUC**	**Precision**	**Recall**	**F1 score**	**Accuracy**	**MCC**
Whole Dataset	0.269 ±0.004	0.885 ±0.002	0.848 ±0.004	0.937 ±0.009	0.890 ±0.002	0.885 ±0.002	0.773 ±0.005
Fine-Tune (lr=0.001)	0.274 ±0.021	0.878 ±0.017	0.860 ±0.012	0.903 ±0.037	0.880 ±0.019	0.878 ±0.017	0.758 ±0.034

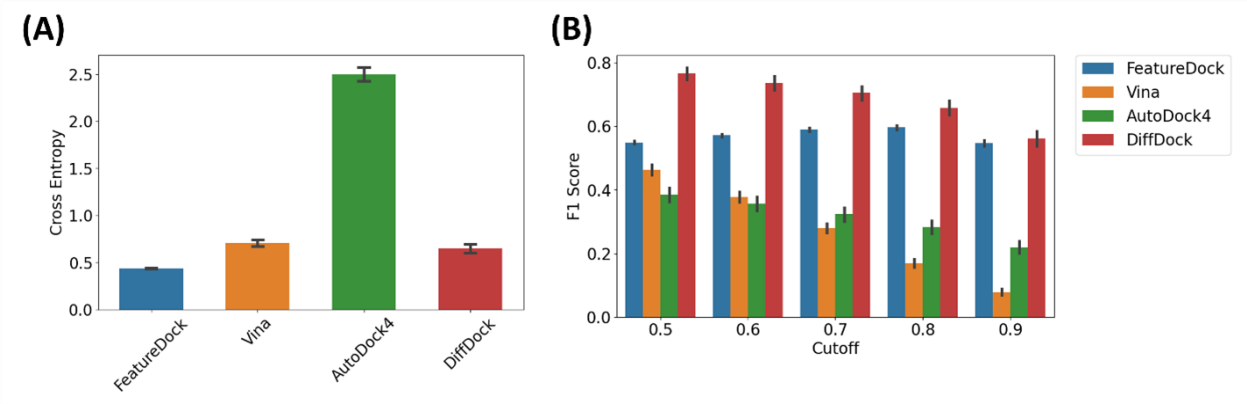


Figure 4.10: Comparison between FeatureDock, Vina, AutoDock4 and DiffDock on PDB-Bind v2020 refined dataset. (A) Cross entropy measures across different methods. (B) F1 Score measures across different methods.

exceeded Vina and AutoDock4 and converged with DiffDock at the most stringent threshold (0.9), as illustrated in Figure 4.10B. Collectively, these results confirm that FeatureDock produces robust and discriminative probability-density envelopes that rival state-of-the-art docking techniques.

4.4.4 Visualization of predicted results

We assessed the spatial interpretability of FeatureDock's predictions using a CDK2-specific case study. FeatureDock generates a probability density envelope by collecting the predicted binding probabilities of all grid points within the query pocket shown in Figure 4.11 which covers ATP-competitive binding regions. The resulting probability-density envelope highlights regions more likely to accommodate ligand atoms with darker color (Figure 4.13C).

Especially, the probability envelopes are sensitive to conformational changes, promising for structure-based drug discovery. Shown in Figure 4.12, the probability density envelope of the inactive and active state of CDK2 revealed significant divergence near Tyr15 in the glycine-rich loop (residues 11–16). This suggests that an allosteric ligand occupying this region could impede the conformational shift required for activation, aligning with the experimentally characterized ANS binding site[282] situated between the ATP pocket and the C-helix which can inhibit cyclin binding.

In general, across 18 pre-aligned CDK2 complexes including both inactive and active conformations, grid points assigned probabilities > 0.8 are located 0.62Å from the true ligand heavy-atom positions (Figure 4.13B), demonstrating high spatial fidelity.

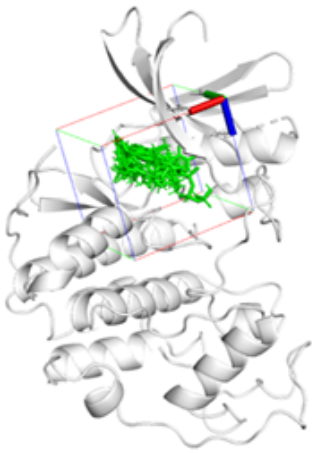


Figure 4.11: Query box and ligands in cocrystal structures. We chose a query box of 18Å $\times$ 16Å $\times$ 16Å box centered at [1.38, 26.15, 9.40][5], which covers the ATP-binding pocket. The representative protein structure shown in the plot is 1B38. Ligands are from 18 pre-aligned complexes. 11 ligands of inactive CDK2 cocrystal structures: 1B38, 1E1X, 1JSV, 1PXO, 1PXP, 2FVD, 2XMY, 2XNB, 5JQ5, 6GUH, 6GUK. 7 ligands of CDK2/Cyclin cocrystal structures: 4BCK, 4BCM, 4BCN, 4BCO, 4BCP, 4IZ3, 6GUE. Ligands in these structures are used to calculate RMSDs of pose prediction in the Section 4.5.2.

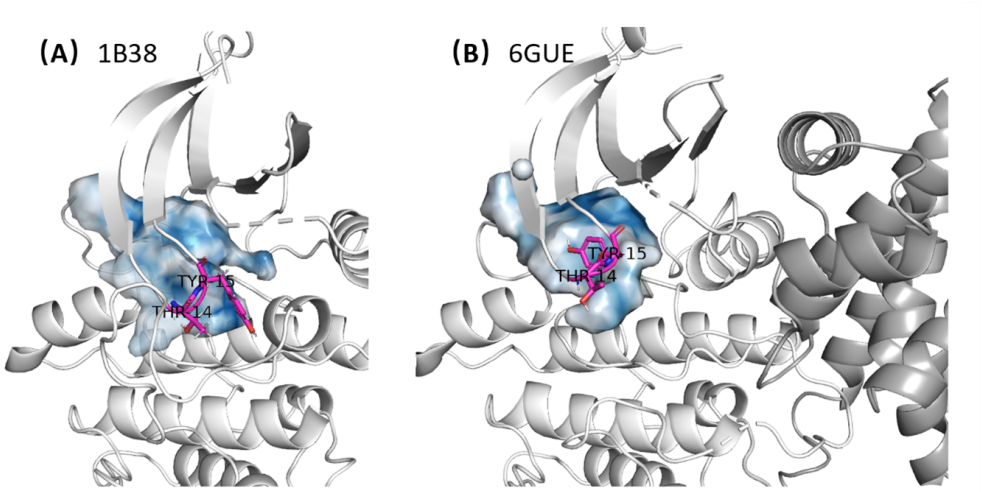


Figure 4.12: Probability maps of two different CDK2 conformations. (A) An inactivated form of CDK2 structure: 1B38. (B) An activated form of CDK2 in complex with Cyclin A: 6GUE. The grid points with probability above 0.8 are plot as surface, darker regions showing higher probabilities. Thr14 and Tyr15 are highlighted in magenta sticks.

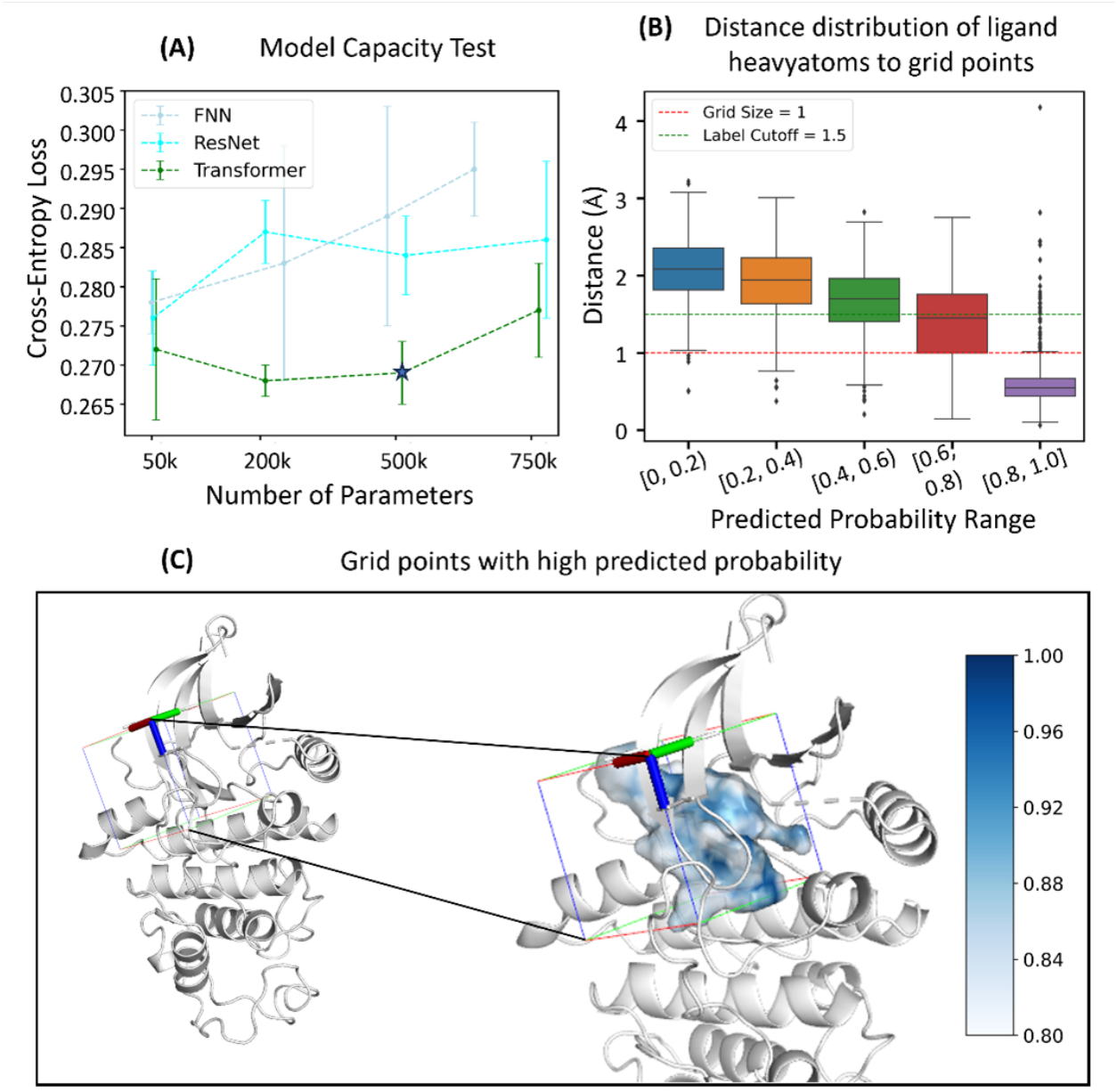


Figure 4.13: Performance and explainability of CDK2 predictions. (A) Comparisons of cross-entropy loss of the validation set of three different architectures: FNN, ResNet, and Transformer. (B) Distance distributions of true ligand atoms to grid points. (C) Query box and the probability density envelope of 1B38 (inactive CDK2). The darker blue regions have higher probabilities.

4.4.5 Explainable AI: Identifications of chemical features contributing most to ligand binding

Although deep learning models are ubiquitous in complex pattern recognition tasks, their adoption in high-stakes domains such as drug discovery, medical imaging, and finance may be constrained by concerns over their limited interpretability. Explainable Artificial Intelligence (XAI)[283] becomes an emerging field as it offers principled techniques that expose the internal logic of these black-box neural networks, fostering scientific insight, regulatory compliance and user trust. Many methods have been proposed to provide *post-hoc* explanations (analyze the model after training) for deep neural networks.

Local Interpretable Model-Agnostic Explanations (LIME)[182, 284] explains the prediction of any classifier by perturbing the input locally, and learning a simple surrogate model to interpret importance from coefficients, thus providing model-agnostic explanations. It is particularly useful for image classifiers. However, each explanation entails hundreds of additional model evaluations, and fidelity depends on the radius of the sampled neighbourhood.

Gradient-based methods compute derivatives of the output with respect to the input, thus more efficient. Saliency maps[285] use raw gradients with respect to the input image and produce pixel-level heat maps. Gradient-weighted Class Activation Mapping (Grad-CAM)[286] is a model-specific method for any CNN-based model which utilizes gradients flowing to the last convolutional layer to produce a coarse localization map, achieving better robustness than the vanilla gradients in Saliency maps. Integrated gradients (IntGrad)[287] was introduced to mitigate the gradient saturation issue that may happen in most gradient-based methods[288].

Attention-based XAI[289] utilizes the attention mechanism that lies at the heart of Transformer architectures. Especially, the normalized attention score matrix offers a built-in mechanism that scores pairwise token interactions. Because these scores are human-readable probabilities, many studies visualize them as explanations[290]. In protein language models, high attention (HA) sites provide guidance to identify critical residues and biological functions[291]. Attention-based explanations are computationally efficient and intuitively appealing, but their validity are still debated[292, 293] and must be verified to ensure that high attention truly implies functional importance.

Owing to the use of physicochemical features as input, the Transformer-based FeatureDock framework produces a separate attention map for each of the 80 physicochemical descriptors by leveraging the interpretability advantages of self-attention. Each map quantifies how strongly a given descriptor at a specific grid point influences the predicted ligand-

occupation probability of itself, effectively yielding a high-resolution, descriptor-specific contribution landscape. Figure 4.14 showcases this strategy for the inactive conformation of CDK2 (PDB ID: 1B38). Figure 4.14B visualizes the spatial attention corresponding to hydrophobicity and the presence of polar residues.

Figure 4.14A aggregates these weights at the residue level. For the hydrophobicity, the aromatic residues Phe82 and Phe80 emerge as the dominant contributors, with hydrophobicity accounting for 2.6% and 2.0% of its binding, respectively. This result is consistent with the result achieved by the pharmacophore analysis of 124 CDK2 co-crystals[294], where the same phenylalanines repeatedly served as hydrophobic anchors. For the polar residues, the highest scores concentrate on Gln85 and the contiguous Gln131–Asn132 region. The polarity of Gln85 contributes 3.9% to its binding. This is consistent with previous MD studies[295, 296] in which Gln85 has been singled out as a selectivity hotspot, supported by the fact that it enhances selectivity for CDK2 over CDK7 by introducing electronegative substituents toward the Gln85–Lys89 region. Although Gln131 and Asn132 have low occurrence in the existing pharmacophore model[294], their high polarity attention weights suggest that engineering ligands capable of hydrogen-bonding to this pair could further improve ligand selectivity over other CDKs.

4.5 Applications in virtual screening and pose prediction

4.5.1 Scoring functions and virtual screening

In virtual screening of the FeatureDock workflow, each candidate ligand is scored by how well its atoms occupy high-probability regions of the query binding pocket. Let $\mathbf{p}$ denote the Cartesian coordinates of N grid points in the potential binding pocket, $\mathbf{P}(\mathbf{p})$ denote the predicted probabilities of the grid points, and $\mathbf{q}$ denote the compound conformer of M heavy atoms. For any *apo* protein structure, $\mathbf{p}$ and $\mathbf{P}(\mathbf{p})$ are fixed in the scoring function, while the rigid-body transformation matrix $\mathbf{T}$ that incorporates translational and rotational operations in the three-dimensional Euclidean space is optimized to pose the ligand $\mathbf{q}$ for the best fit. FeatureDock defines a scoring function that averages the probability-weighted contributions of neighbouring grid points (within 1.5Å) for each ligand atom. Additionally, an objective function aggregates over all grid points inside a broad probability envelope (default cutoff 0.50). Gradient-based local optimization using L-BFGS-B algorithm iteratively updates $\mathbf{T}$ toward better fits, thereby aligning the ligand to high-probability regions in the predicted envelope.

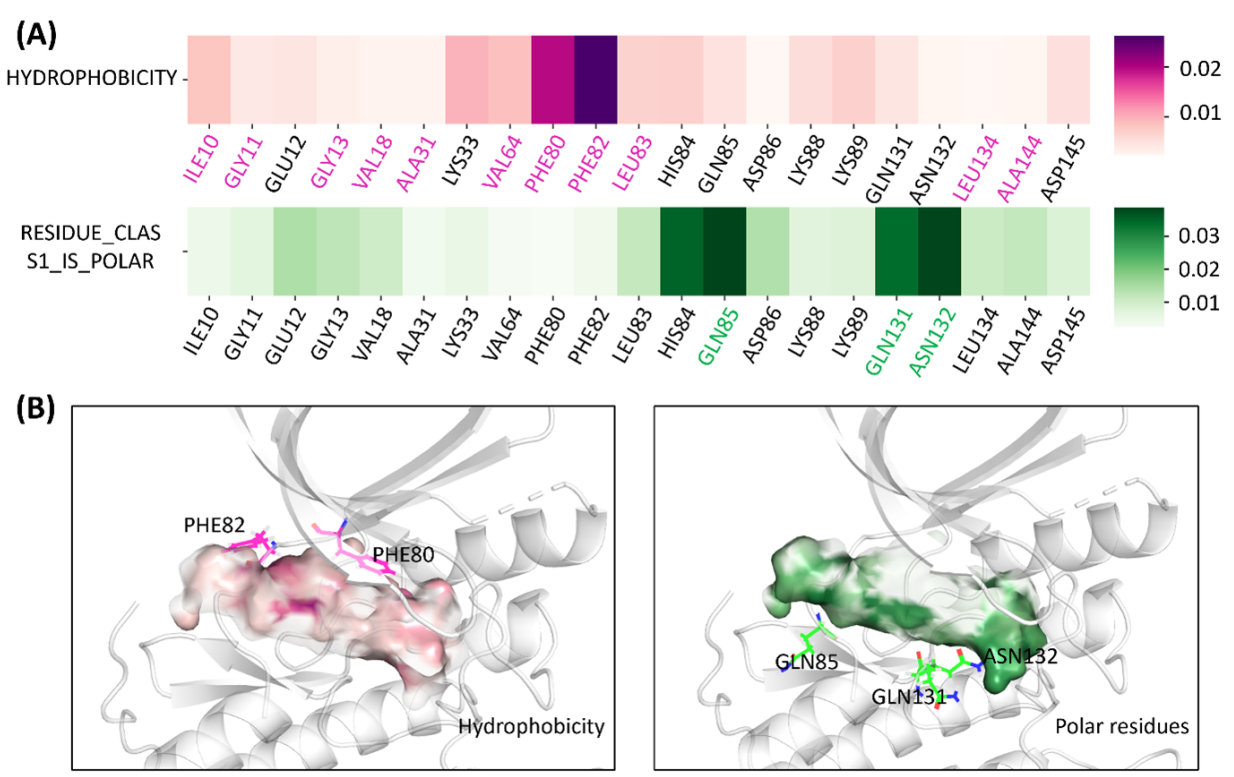


Figure 4.14: Model interpretability visualized by attention weight of different input properties. (A) Heatmap of property contribution to the amino acids of interest by averaging attention weights of grid points around each amino acid. The picked residues frequently form contact with ligands in cocrystal structures. The text of hydrophobic residues and polar residues are colored in magenta and green, respectively. The range of attention weights is $[0, 1]$. (B) The attention weight map of different properties of the spatial grid points. Regions in darker color have higher contribution values in the attention analysis.

$$\begin{aligned} \text{Scoring Function} &= \frac{1}{|\mathbf{q}|}\sum_{\mathbf{q}_j}\frac{1}{|N_p(T\mathbf{q}_j)|}\sum_{\mathbf{p}_i\in N_p(T\mathbf{q}_j)}\mathbf{P}(\mathbf{p}_i)e^{-\|\mathbf{p}_i-T\mathbf{q}_j\|^2} \\ \text{Objective Function} &= \frac{1}{|\mathbf{q}|}\sum_{\mathbf{q}_j}\frac{1}{|\mathbf{p}|}\sum_{\mathbf{p}_i}\mathbf{P}(\mathbf{p}_i)e^{-\|\mathbf{p}_i-T\mathbf{q}_j\|^2} \end{aligned} \tag{4.7}$$

To ensure exhaustive sampling, up to 20 conformers are generated for each molecule using RDKit, guided by torsion angle and medicinal chemistry priors (`useExpTorsionAnglePrefs=True`, `useBasicKnowledge=True`). Conformers are clustered at an RMSD threshold of 1.0Å using DBSCAN to eliminate redundancy. Each conformer undergoes up to 500 independent random rotations about the pocket center, producing a maximum of 10,000 optimization trajectories per compound. These are subsequently clustered using DBSCAN to identify

distinct poses. The full sampling and clustering procedure is repeated across three random seeds to mitigate stochastic artifacts.

4.5.2 Preparation of query structures and compound libraries

To evaluate the performance of virtual screening, two medicinally relevant systems (inactive CDK2 and ACE) were selected. Candidate compounds were drawn from ChEMBL[279] bioassay experiments of each protein and split by potency thresholds where compounds with $IC_{50} < 10nM$ are considered as strong inhibitors and compounds with $IC_{50} > 10\mu M$ are considered as weak inhibitors.

The filtered compound library from CHEMBL demonstrated a low compound similarity to ligands in PDBBind refined dataset (Figure 4.15A). All molecules with $> 95\%$ ECFP4 Tanimoto similarity to any ligand in the PDBBind refined set were removed, thus simulating the discovery of genuinely novel chemotypes from a pre-selected drug-like compound library.

The CDK2 query structure (PDB ID: 1B38) was used in conjunction with a pocket grid defined by residues within 5Å of 11 pre-aligned ligands from inactive CDK2 cocrystals (Figure 4.16A). Additional library filters were applied to the bioassay results from CHEMBL301 and discarded compounds exceeding 400Da. The compounds explicitly annotated as CDK2/cyclin inhibitors are further removed because the probability density envelope is conformational sensitive. The above filtering results in a 147-compound library whose physicochemical distributions are balanced across potency classes (Figure 4.16B).

For angiotensin-converting enzyme (ACE), the *apo* structure 3BKL was employed. The pocket grid encompassed occupied regions of ligands from four pre-aligned structures (2OC2, 3BKK, 3BKL, 6F9U) following the same cutoff of 5Å (Figure 4.17A). The ACE library, sourced from ChEMBL1808, retained only compounds between 400 and 600 Da and compliant with standard drug-likeness heuristics (logP$\leqslant$ 5, hydrogen donors$\leqslant$ 5, hydrogen acceptors $\leqslant 10$, number of rotatable bonds $\leqslant 5$), yielding 94-compound library for screening (Figure 4.17B) with balanced properties.

4.5.3 Parameters used in the docking programs

For benchmarking, standard protocols were applied to classical docking tools. Proteins were prepared using AutoDockTools 1.5.7 and converted to PDBQT format. Docking boxes were defined as $18 \times 16 \times 16$ Å^3 for CDK2 (center: $[1.38, 26.15, 9.40]$) and $18 \times 22 \times 24$ Å^3 for ACE (center: $[43, 45, 44]$). AutoDock Vina and Smina were run with an exhaustiveness of

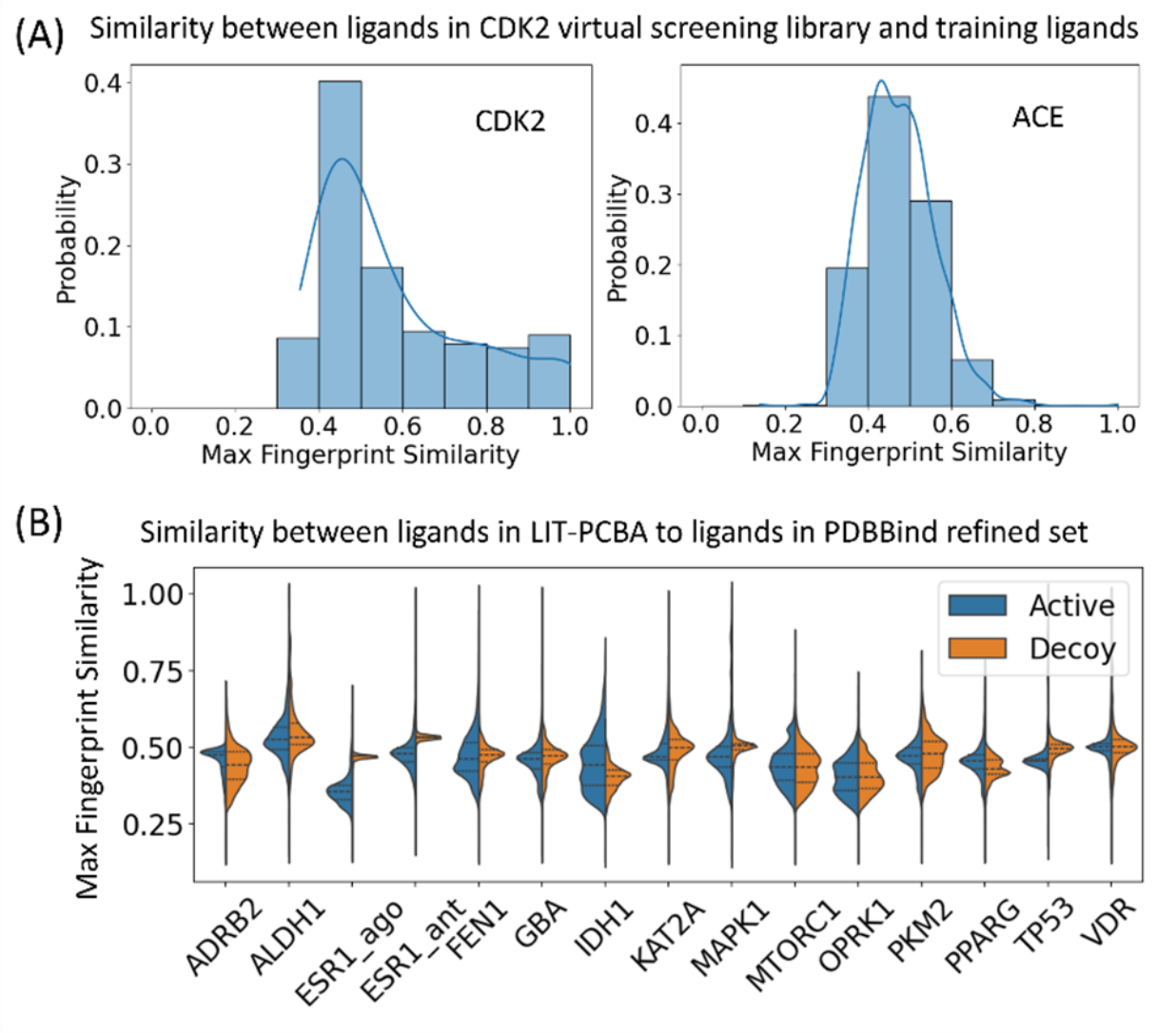


Figure 4.15: Similarity distribution of compounds across benchmark datasets and ligands from the PDBBind v2020 refined dataset. (A) Similarity distribution of compounds in virtual screening libraries and ligands in training from PDBBind v2020 refined dataset. (B) Similarity distribution of compounds in the LIT-PCBA[6] benchmark dataset and ligands from PDBBind v2020 refined dataset.

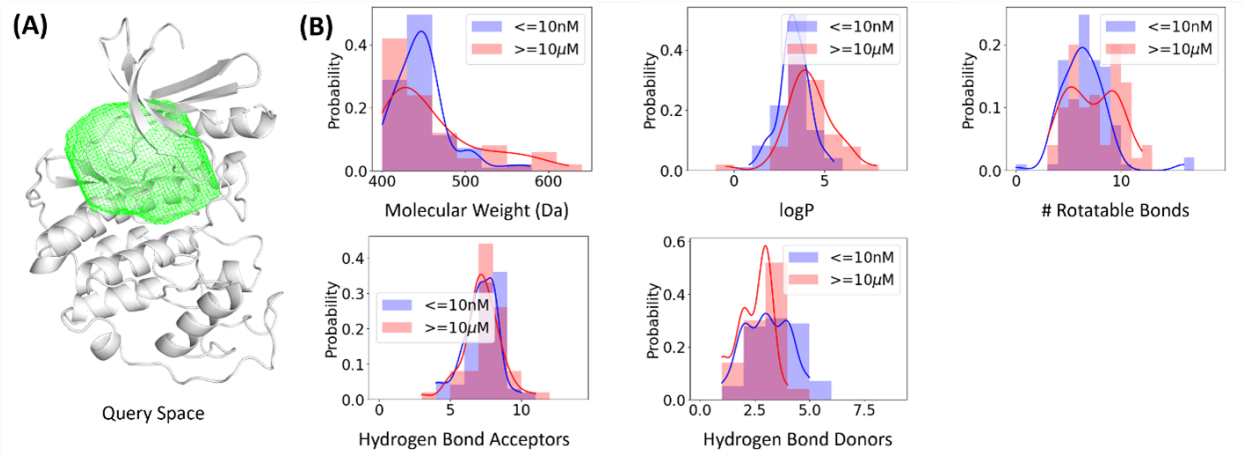


Figure 4.16: Virtual screening setup for an inactivated form of CDK2. (A) Query space used in the inactivated form of CDK2 structure (PDBID: 1B38). (B) Molecular properties of compounds in the 147-compound CDK2 virtual screening library.

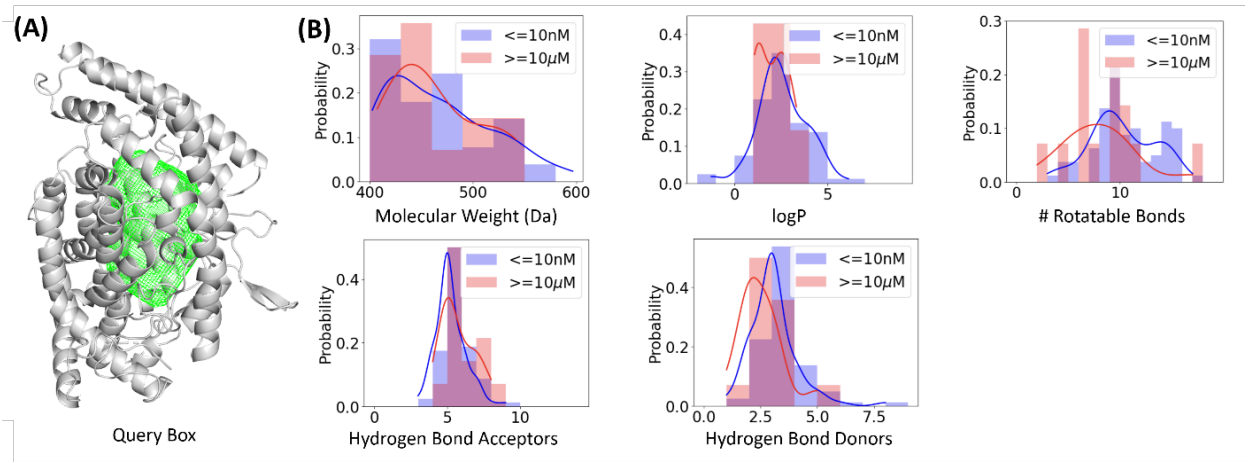


Figure 4.17: Virtual screening setup for ACE. (A) Query space used in the ACE structure (PDBID: 3BKL). (B) Molecular properties of compounds in the 94-compound ACE virtual screening library.

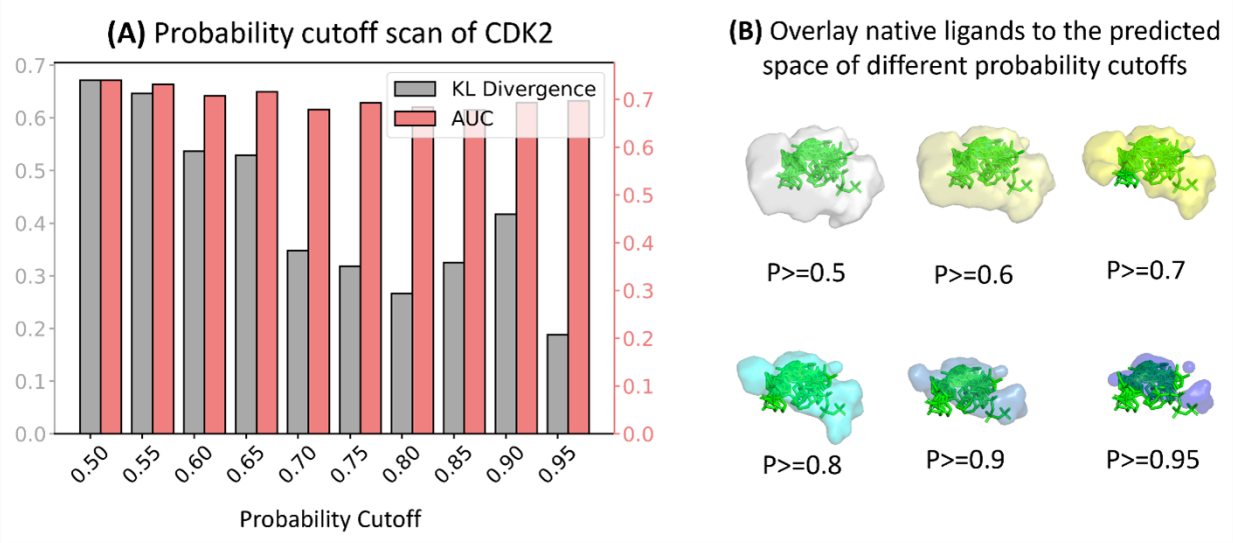


Figure 4.18: Hyperparameter scan of probability cutoffs for inactive CDK2. (A) KL Divergence and AUC values under different probability cutoffs. According to the scanning result, we reported p>=0.50 as the best criteria in the scoring function to distinguish strong inhibitors from weak inhibitors for the CDK2 pocket. (B) Probability density envelopes under different probability cutoffs.

8 and returned the top 20 poses per compound. DiffDock was executed using its published default settings (20 inference steps, 40 samples per complex).

4.5.4 Hyperparameter scanning in FeatureDock virtual screening

Selecting an optimal probability threshold is essential for the scoring function of FeatureDock because the cutoff value directly governs which regions of the predicted envelope contribute to the final score. The recommended procedure is to treat the cutoff as a tunable variable and scan a series of cutoffs against available bioassay data, retaining the threshold that maximizes the area under the ROC curve (AUC) to distinguish strong ($IC_{50} < 10nM$) from weak ($IC_{50} > 10\mu M$) inhibitors. For ACE, a stringent cutoff of $p \geqslant 0.90$ yielded the highest AUC and KL-divergence, consistent with its structurally rigid active site (Figure 4.19). In contrast, the more flexible CDK2 pocket achieved best results at $p \geqslant 0.50$ (Figure 4.18). These findings are further corroborated by crystallographic B-factor analyses (Figure 4.20), as ACE exhibits lower conformational variability, warranting a higher threshold. In general, a default threshold of $p \geqslant 0.90$ is recommended when activity data are unavailable.

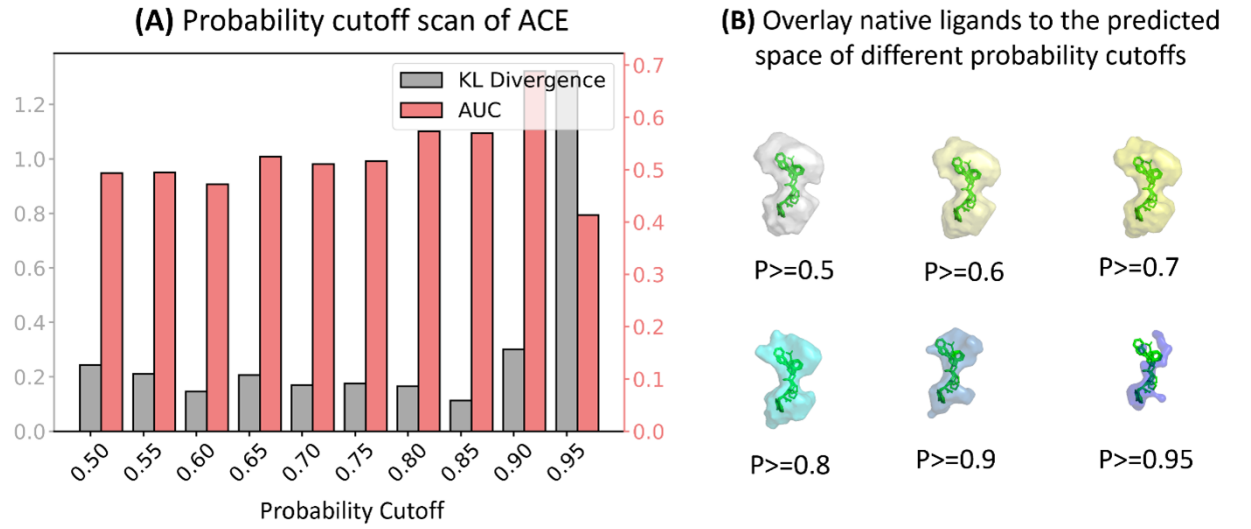


Figure 4.19: Hyperparameter scan of probability cutoffs for ACE. (A) KL Divergence and AUC values under different probability cutoffs. We reported p>=0.90 as the best criteria in the scoring function to distinguish strong inhibitors from weak inhibitors for the ACE pocket. (B) Probability density envelopes under different probability cutoffs.

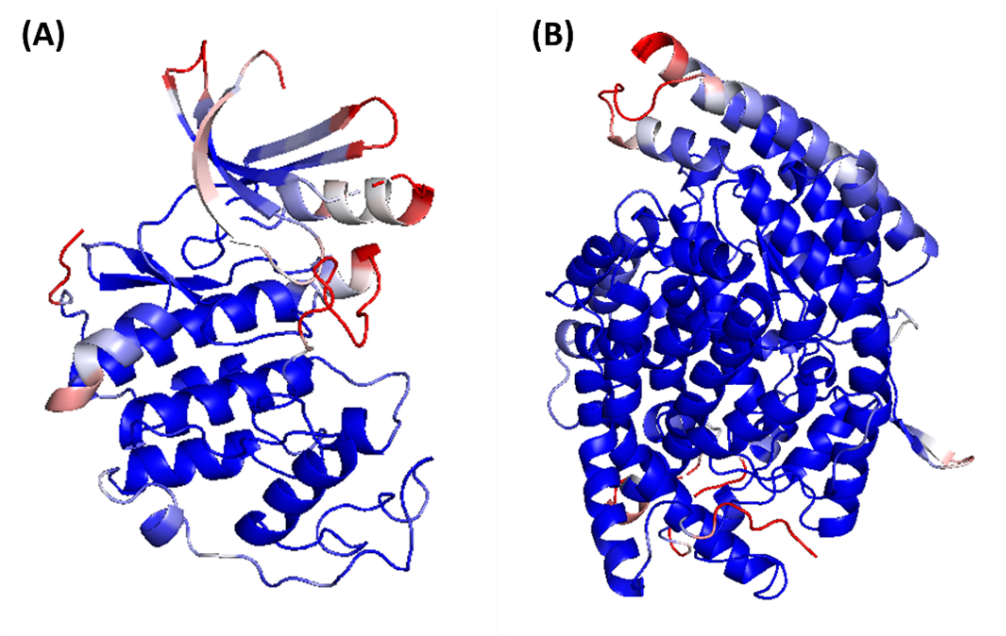


Figure 4.20: Protein flexibility. Coloring protein flexibility based on B-factors. PyMol employs a color spectrum ranging from blue to white to red (low to high), with the minimum of B-factor colorbar set at 20 and the maximum at 50. (A) B-factors of an inactivated form of CDK2 (PDBID: 1B38). (B) B-factors of ACE (PDBID: 3BKL).

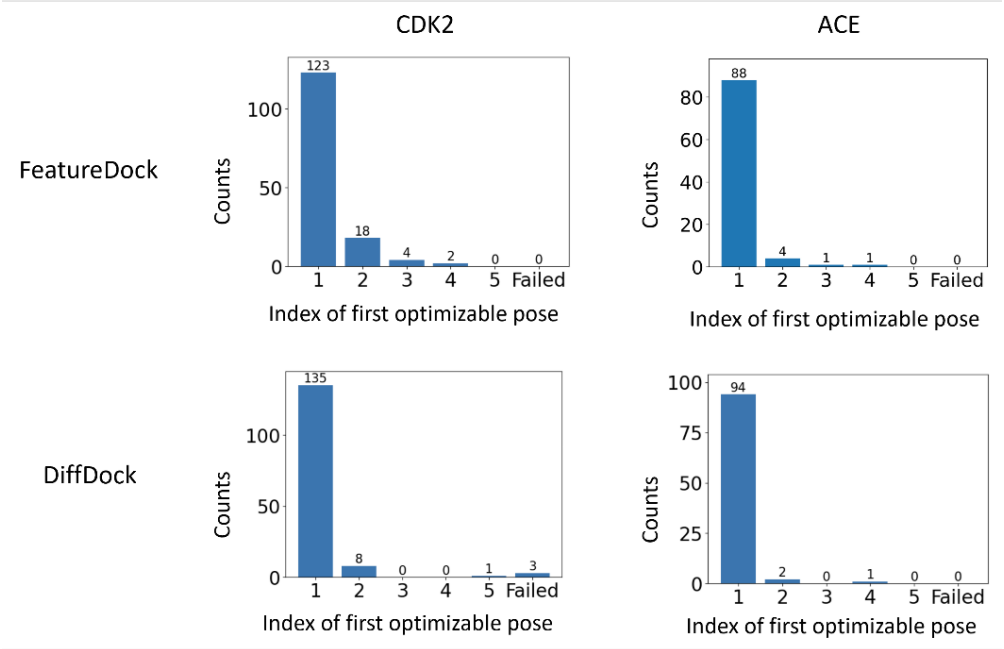


Figure 4.21: Index of first optimizable predicted poses when using AutoDock Vina affinity smaller than 0 as a threshold. DiffDock failed to predict optimizable poses in its top-5 predictions for three compounds from the 147-compound CDK2 virtual screening library. These three compounds are not included in the evaluation results of KL divergence and AUC for DiffDock in Figure 4.17

4.5.5 Post-validation of deep learning docking methods

As deep learning docking methods do not inherently enforce physical constraints[297], steric clashes may arise. To ensure geometric plausibility, all predicted poses from FeatureDock and DiffDock were post-validated via local energy minimization using AutoDock Vina. Poses with refined affinity > 0 kcal/mol were considered unphysical and discarded. This procedure successfully identified and filtered out several problematic DiffDock predictions (Figure 4.21), thereby improving the reliability of evaluation metrics. Future versions of FeatureDock may incorporate van der Waals terms directly into the optimization objective to obviate this *post-hoc* validation.

4.5.6 Evaluation metrics in virtual screening

Scoring performance was assessed using the Kullback-Leibler (KL) divergence between the score distributions of strong and weak inhibitors and by the AUC under the receiver operating characteristic curve. An higher KL divergence indicates sharper discrimination between classes, while a larger AUC reflects better ranking of potent binders.

4.5.7 FeatureDock outperforms DiffDock, Smina and AutoDock Vina in differentiating strong and weak inhibitors

4.5.7.1 Inactive CDK2

In the 147-compound screening library for inactive CDK2, shown in Figure 4.22A and B, FeatureDock delivered a KL divergence of 0.67, outperforming DiffDock (0.39) and the near-zero separations obtained with Vina and Smina (0.04 each). Moreover, AUC results similarly favored FeatureDock (0.74) and DiffDock (0.76) over Vina and Smina (0.43). Visual inspection supports these findings. The nanomolar inhibitor CHEMBL402158 positions its 1H-indazole core entirely within the $p \geqslant 0.95$ regions adjacent to Phe80 and Phe82, and the rest of the scaffold occupies contiguous high probability regions with $0.90 \leqslant p < 0.95$, aligning well with its experimental potency of $7 n M$. Conversely, the micromolar compound CHEMBL3421971 passed a lower probability regions due to the linker connecting its benzene and piperazine rings due to the geometry constraints, lowering its score in line with a measured IC_{50} of $13.7 \mu M$. CHEMBL3798066 achieved even lower score because this compound occupied regions with even lower probabilities. Compound poses overlaied on the probability density envelope provide an intuitive way to visualize and explain the protein-ligand binding interactions.

4.5.7.2 ACE receptor

The screening for ACE demanded a more restrictive scoring threshold because the pocket is known to be rigid, as demonstrated by the higher crystallographic B-factors of CDK2 (Figure 4.20). Hyperparameter scanning on bioassay data identified 0.90 as the optimum probability cut-off for rescoring (Figure 4.19) as it achieves the highest AUC.

Under this stricter envelope, FeatureDock differentiated actives from inactives decisively (KL = 0.30, AUC = 0.69), outperforming the other three methods (Figure 4.23A and B). The top-ranked compound, CHEMBL100413, nests its carboxylate oriented to the active site and forms a hydrogen bond with Tyr520 which is an experimentally validated binding residue[298, 299] as observed in potent ACE inhibitors. Inactive controls in Figure 4.23C achieved lower scores because they either protruded the probability envelope formed by grid points with $p \geqslant 0.80$ (CHEMBL5220968) or missed the optimal regions (CHEMBL5177969) entirely.

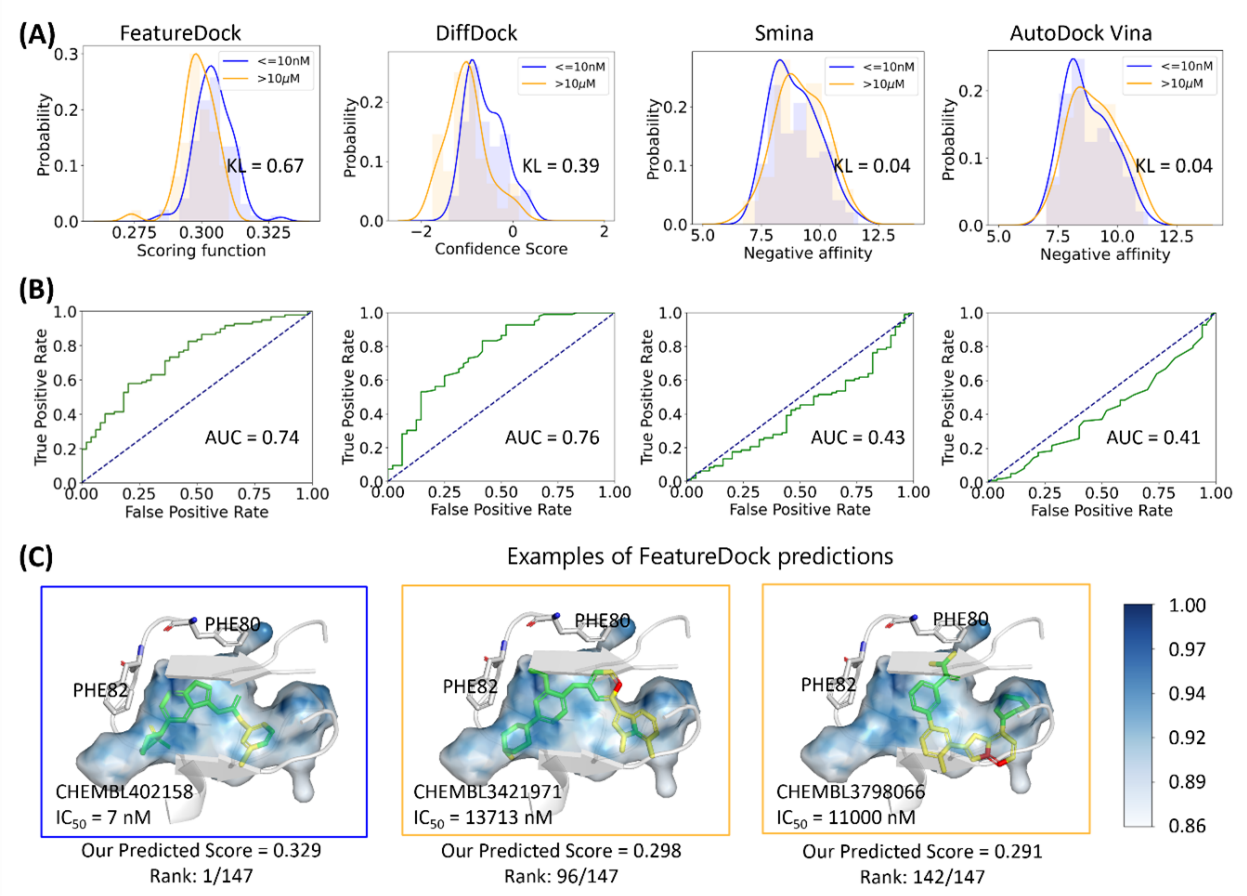


Figure 4.22: Probability density envelope guided virtual screening of an inactivated form of CDK2. (A) KL divergence of score distribution on strong inhibitors and weak inhibitors from a filtered CDK2 bioassay dataset which contains 147 compounds. (B) AUC of scoring functions on the 147-compound library. (C) True positive (the blue boxed) and true negative (the orange boxed) examples of predicted poses overlaid with the predicted probability density envelope. The colorbar represents colors of different probability values. Compound atoms in the regions of p >= 0.95, 0.90 <= p < 0.95 and 0.80 <= p < 0.90 are colored in green, yellow, and red respectively. Surrounding protein structures are shown in white ribbon, with important binding residues Phe80 and Phe82 highlighted in white sticks.

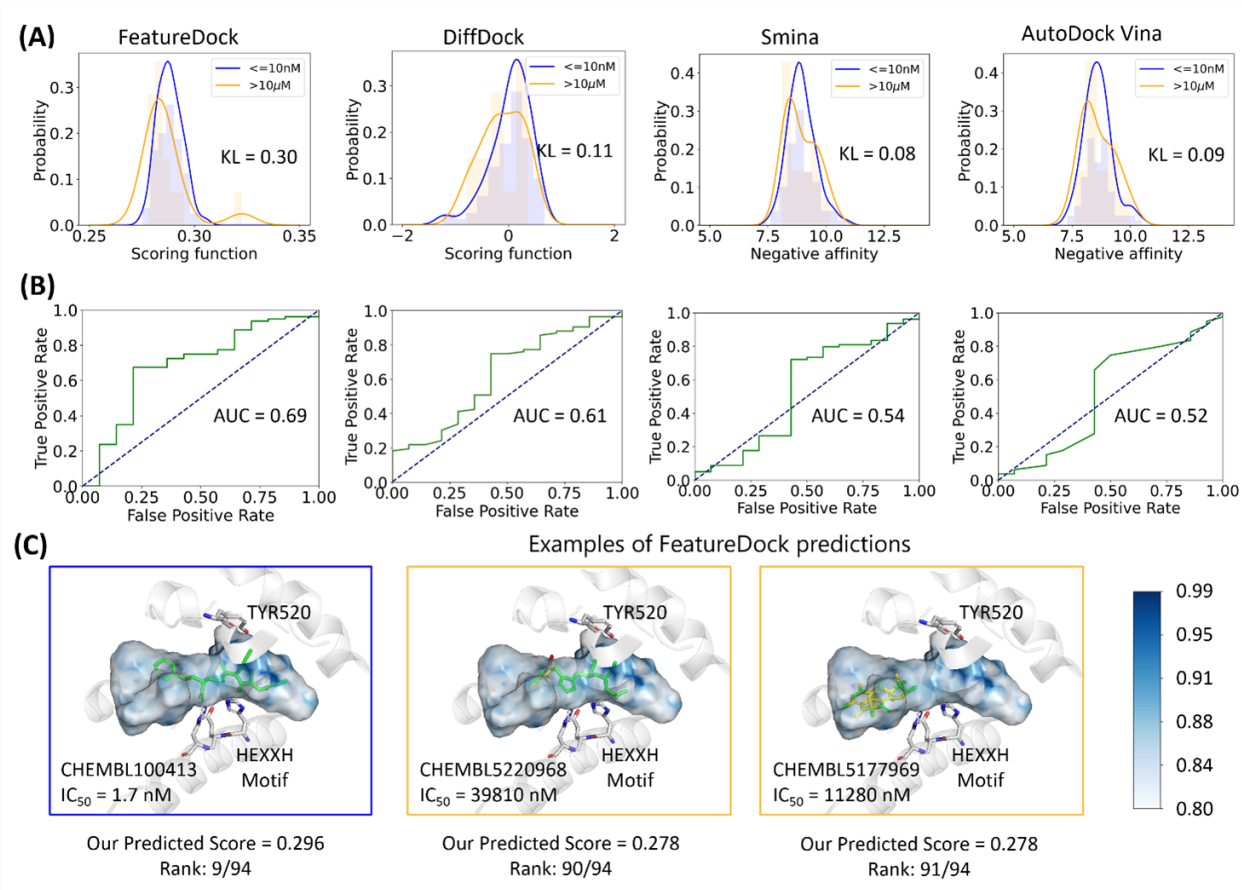


Figure 4.23: Probability density envelope guided virtual screening of ACE. (A) KL divergence of score distribution on strong inhibitors and weak inhibitors from a filtered ACE bioassay dataset which contains 94 compounds. (B) AUC of scoring functions on the 94-compound library. (C) True positive (the blue boxed) and true negative (the yellow boxed) examples of predicted poses overlaid with the predicted probability density envelope. Compound atoms in the regions of p >= 0.95, 0.90 <= p < 0.95 and p <= 0.80 are colored in green, yellow, and red respectively. The colorbar represents colors of different probability values. Surrounding protein structures are shown in white ribbon. The active site (NEXXH motif) and the Tyr520 residue are shown as white sticks.

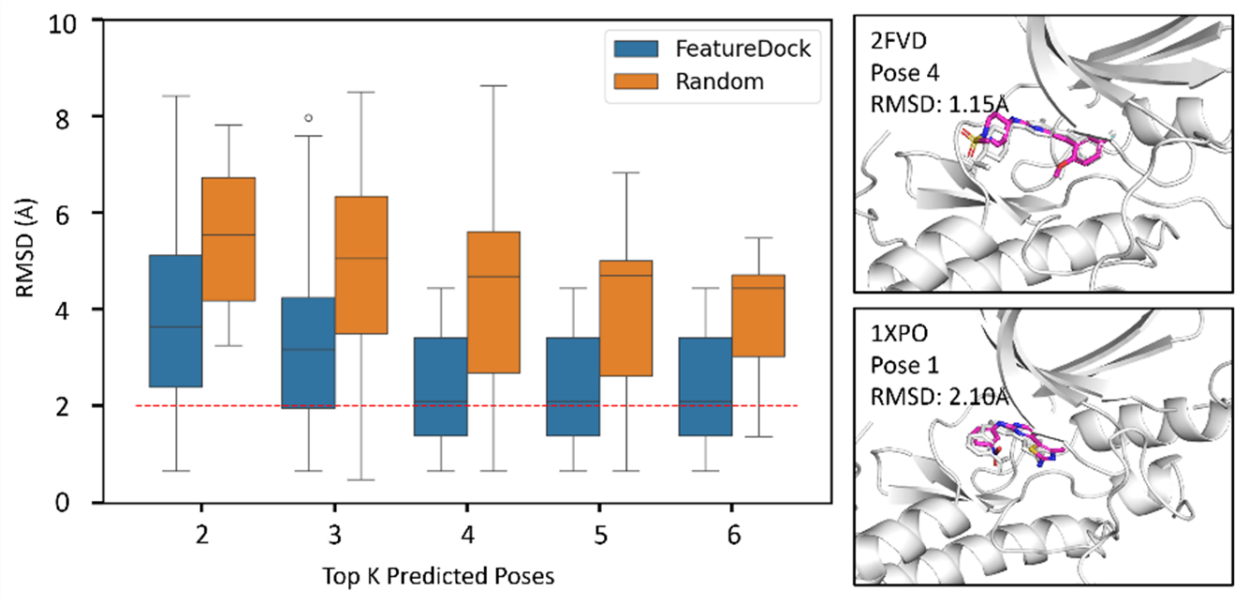


Figure 4.24: Pose prediction from FeatureDock on CDK2. The query space of each protein refined by protein residues within $[1, 6]$Å away from the heavy atoms of its native ligand. The native rotamer was used in the posing process. Left panel: Distribution of root mean square distances (RMSDs) between predicted poses and native poses. Right panel: Examples of predicted poses. The ground truths and the predictions are presented in white sticks and magenta sticks, respectively.

4.5.8 FeatureDock correctly finds the binding pose of CDK2 inhibitors

Pose accuracy was assessed on 11 native ligands of inactive CDK2 with available co-crystal structures. Considering the four highest-scoring poses per ligand, FeatureDock achieved an average RMSD of 2.4Å and a median of 2.1Å relative to the ground true poses (Figure 4.24). Although the conventional success threshold in docking benchmarks is an RMSD of 2.0Å, these results are still notable because the current posing algorithm only relies on maximizing the probability occupancy and does not contain any explicit physics-based interaction terms.

4.6 Scoring power comparison among FeatureDock, traditional docking, and machine-learning based scoring functions

4.6.1 Scoring performance

We benchmarked the scoring performance of FeatureDock against RF-Score v1[274] using protein-ligand complexes from the PDBBind v2020 refined set that contain drug-like ligands ($400\ \text{Da} \leqslant \text{MW} \leqslant 600\ \text{Da}$). When RF-Score is evaluated on complexes that were also present in its training set, it displays an almost perfect linear relationship with the experimental binding affinities (Pearson correlation=0.953, Figure 4.26A). However, on complexes that were excluded from training, the correlation decreases markedly (0.586, Figure 4.26B).

FeatureDock, whose model is not trained on affinity labels, attains a lower correlation (0.408, Figure 4.25A) but does so without explicit knowledge of experimental affinities. For further comparison, we evaluated RF-Score-VS[300], a variant of RF-Score but trained on the DUD-E dataset[301], which produced an even lower correlation (0.343; Figure 4.26C), highlighting poor cross-dataset generalization of RF-Score descriptors and algorithms.

These findings suggest that while affinity-supervised methods such as RF-Score can attain high accuracy within their training domain, their transferability to new datasets is limited. FeatureDock, by contrast, maintains consistent accuracy across novel targets and ligands, and its probabilistic scoring framework naturally integrates with its pose-optimization procedure, providing an interpretable and competitive alternative for structure-based virtual screening.

A broader comparison incorporating traditional empirical scoring (AutoDock Vina[260] and Vinardo[264]), a deep-learning-based re-scoring model (GNINA CNNaffinity[265]), and FeatureDock is summarized in Figure 4.25. Among the five methods, GNINA achieves the highest Pearson correlation (0.720) with experimental affinities, followed by RF-Score (0.647). FeatureDock and AutoDock Vina display comparable performance (≈ 0.41), whereas Vinardo exhibits the weakest association (0.394). Notably, FeatureDock reaches a correlation similar to that of Vina even though its scoring function is derived solely from the probability density envelopes predicted from protein local environments, rather than from any direct affinity supervision. This result demonstrates that a physics-inspired, probability-based docking score can approach the accuracy of empirical energy functions while retaining the capacity to generalize beyond the structures used in model training.

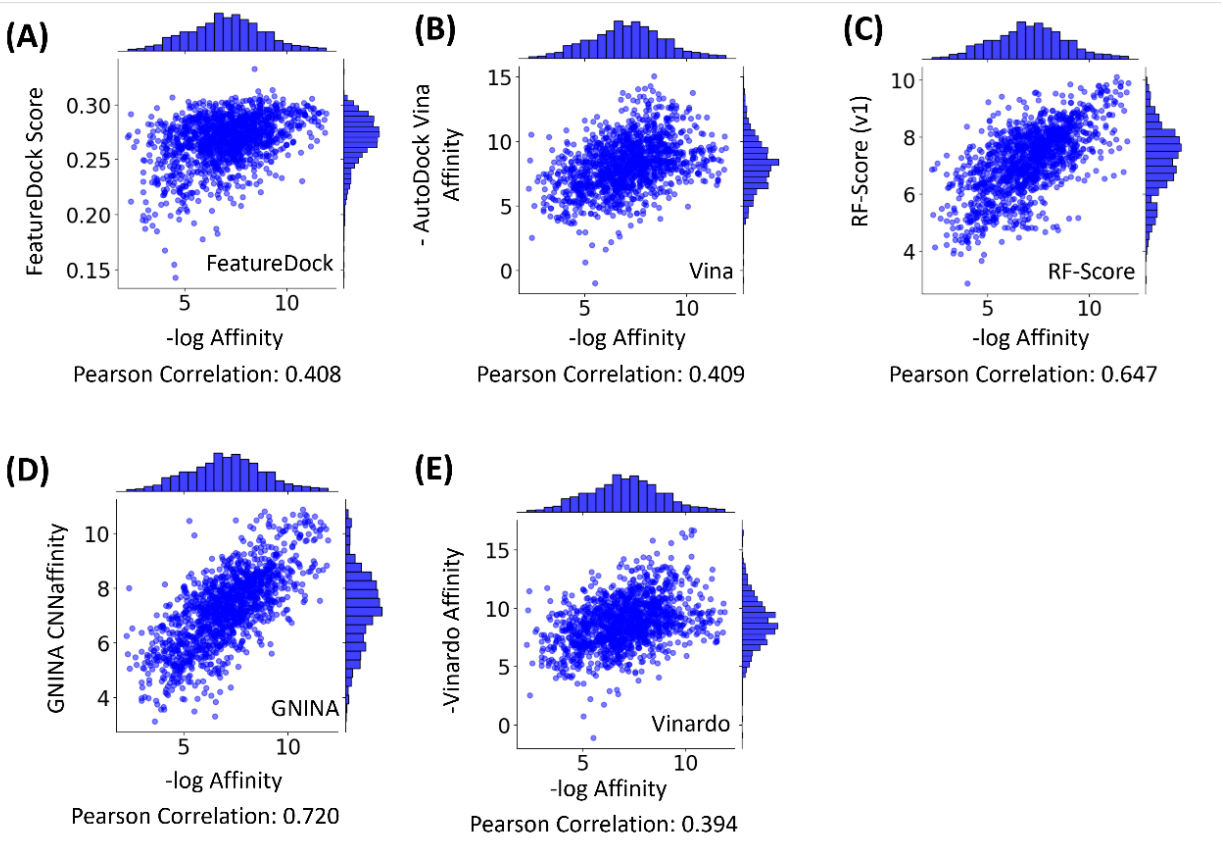


Figure 4.25: Comparison of different scoring functions on ligands with druglike molecular weights (400<=MW<=600) from the PDBBind v2020 refined dataset. (A) FeatureDock Score (trained on PDBBind v2020 refined dataset without binding affinity). (B) AutoDock Vina Affinity (Scoring function derived from the PDBBind dataset). (C) RF-Score v1 (trained on PDBBind v2007 refined-without-core dataset). (D) Gnina CNNaffinity (trained on PDBBind v2019 dataset and cross-docking dataset). (E) Vinardo affinity

4.6.2 Scoring speed evaluation

The practical utility of a docking platform depends not only on scoring accuracy but also on computational throughput. We benchmarked the runtime required to score a single compound using 100 randomly selected protein–ligand complexes from the PDBBind v2020 refined set. All timing experiments were conducted on Intel(R) Core (TM) i7-10510U.

For the scoring phase alone (Figure 4.27 left panel), FeatureDock exhibited the highest cost (~ 90 secs/compound on average). This reflects the cumulative cost of grid point featurization, Transformer model loading, inference over grid points, and score computation. GNINA took ~ 25 secs/compound on average, while AutoDock Vina remained the most time-efficient (under 5 secs).

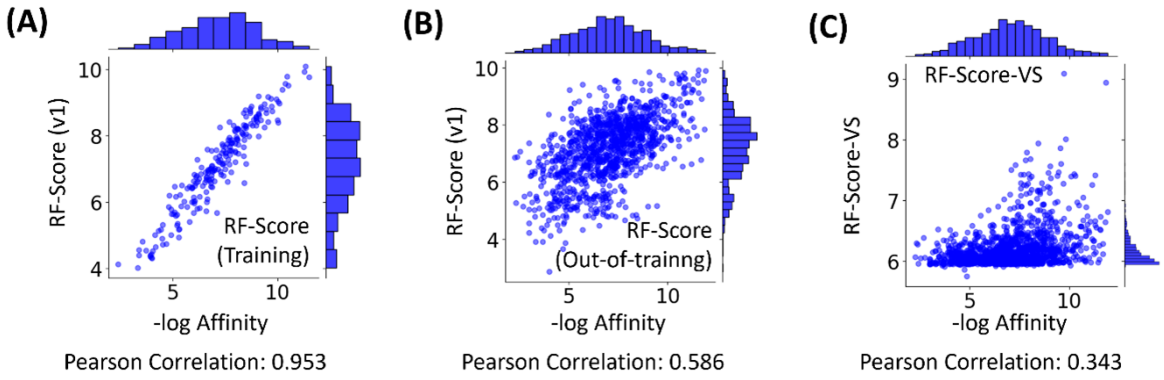


Figure 4.26: The scoring power of RF-Score on ligands with druglike molecular weights (400<=MW<=600) from the PDBBind v2020 refined dataset. (A) Pearson correlation of RF-Score on training samples. (B) Pearson correlation of RF-Score on out-of-training samples. (C) Pearson correlation of RF-Score-VS (trained on DUD-E dataset) on druglike molecules.

Despite its higher computational cost, FeatureDock remains suitable for medium-throughput screening pipelines, particularly when multiple ligands target the same protein structure. In such cases, the predicted probability map can be reused, amortizing the cost across many compounds.

When considering the pose prediction (Figure 4.27 right panel), diffusion-based DiffDock is the slowest method ($\sim$ 1500 secs/compound) because of the diffusion process and lack of hardware acceleration. FeatureDock took $\sim$ 600–700 secs to pose each compound, influenced largely by the number of Monte Carlo trials, conformer diversity, and clustering RMSD thresholds. GNINA ($\sim$ 300 secs/compound) and Vina ($<$ 100 secs/compound) remained the most time-efficient.

It is important to note that the scoring and posing phages are decoupled in the two-step virtual screening strategy adopted by FeatureDock. The current posing step contains a simple Monte-Carlo sampling followed by L-BFGS-B optimization based on the objective function, which should be optimized for efficiency in the future.

4.7 Conclusions and future perspectives

In this chapter, we presented FeatureDock, a unified and feature-centric docking framework that couples a physicochemically informed representation of the protein local environment with a Transformer encoder to predict three-dimensional probability density envelopes.

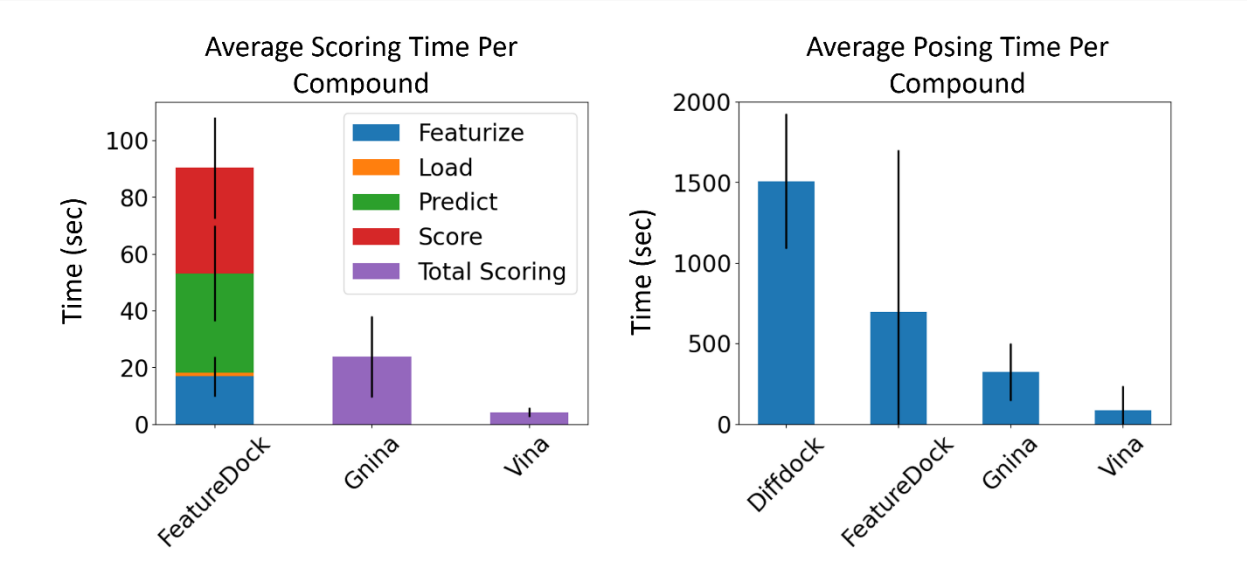


Figure 4.27: Computational efficiency test of FeatureDock. (A) Scoring speed of FeatureDock, AutoDock Vina and GNINAThe total Scoring time of FeatureDock is decomposed to featurize grid points, load Transformer model, predict probabilities of grid points and score the compound based on probabilities. (B) Pose prediction speed of FeatureDock, AutoDock Vina GNINA and DiffDock

Systematic hyperparameter scanning demonstrated that the Transformer encoder delivers markedly higher and more stable predictive accuracy than feed-forward neural networks and residual convolutional architectures under comparable capacity constraints. Moreover, the attention weights yield chemically intuitive contribution maps that reveal how local features of the query protein drive the predicted binding propensity, thus providing insights for rational ligand design.

These predicted probability envelopes quantify the spatial preference of ligand atoms as discrete grid points and probability values, based on which a scoring function and an efficient ligand-psing algorithm was designed for the purpose of virtual screening. When validated on the bioassay compound libraries for the inactive CDK2 and the ACE, FeatureDock surpassed DiffDock, Smina, and AutoDock-Vina in distinguishing high-affinity inhibitors from weak binders, attesting to its robustness and generalizability across distinct protein classes during virtual screening.

A promising future direction is to extend FeatureDock into a generative framework capable of performing one-shot, end-to-end docking by leveraging generative AI models. This approach could learn to output plausible ligand configurations conditioned on the protein pocket characterized by spatial local physicochemical environments.

A natural extension of FeatureDock is to reformulate its training objective within a multi-task learning framework that simultaneously predicts multiple protein–ligand interactions including hydrogen bonds and π-π stacking alongside the existing positional probability densities. Providing these additional interaction signals would encourage the Transformer to encourage the model to learn a richer, more chemically expressive representation of protein pocket, thus improving both pose prediction and affinity ranking. Moreover, the multi-task attention weights would afford more granular and chemically interpretable contribution maps.

Because FeatureDock is trained to reconstruct a generic mapping from local protein features to spatial probability densities, rather than to memorize ligand coordinates or affinity labels, it learns transferable three-dimensional embeddings. These embeddings can serve as a foundation for a variety of downstream tasks, including affinity prediction, binding site identification, and grid-based ligand generation, with minimal additional training. In this regard, FeatureDock exhibits properties aligned with those of a foundation model for protein–ligand interaction, capable of supporting diverse applications across chemical and structural biology.

Furthermore, FeatureDock's probability density envelopes provide a spatially resolved prior that can complement the embeddings of protein language models (PLMs). While PLMs encode sequence-based evolutionary and functional information, they lack explicit surface-level structural descriptors. A multimodal framework combining FeatureDock's structure-aware outputs with PLM-derived representations could yield spatially conditioned, sequence-informed models of ligand binding, advancing the integration of sequence and structure in drug discovery pipelines.

Lastly, the physicochemical descriptors employed by FeatureDock are sensitive enough to backbone and side-chain rearrangements, the framework naturally lends itself to ensemble docking. Generating probability envelopes over a curated set of experimentally determined or MD-derived conformers would adapt this framework for flexible receptors, enabling the identification of allosteric binding sites and informing the design of ligands capable of selective conformational stabilization.

These developments will elevate FeatureDock from a standalone docking engine to an extensible, interpretable and consensus-driven platform for next-generation structure-based drug discovery.

5 Conclusions and Future Perspectives

5.1 Summary of key contributions

This thesis presents two novel deep learning frameworks, ChemPLAN-Net and FeatureDock, developed to advance computational strategies in fragment-based drug discovery (FBDD) and structure-based docking, respectively. These two contributions are unified by their emphasis on integrating physicochemical insights from protein local environments with modern neural network architectures to accelerate early-stage drug design.

ChemPLAN-Net is a ResNeXt-based model that predicts protein-fragment interactions by learning from FEATURE vectors encoding local physicochemical environments of protein functional centers. A key contribution lies in the design of reducing a multi-label, multi-class classification task to a binary classification task by using local environment-fragment fingerprint pairs.

My technical contribution to ChemPLAN-Net is the development of an improved dataset curation pipeline, enabling high-throughput, high-quality training data generation from protein–ligand co-crystal structures. This pipeline ensures consistency, extensibility, and compatibility with different protein classes, positioning ChemPLAN-Net as a flexible foundation for broader applications in FBDD.

To bridge the gap between fragment hits and complete drug-like compounds, a virtual synthesis engine was constructed to filter and screen compounds. It applied Lipinski's RO5 descriptors to enrich drug-like compounds, took ChemPLAN-Net predicted fragments as input to calculate atom coverage scores and find candidates, and finally docked candidates using AutoDock Vina to validate binding affinities.

Fragment linking is another approach utilized to develop full drug-like compounds based on ChemPLAN-Net predicted fragments in the SARS-CoV-2 M^{pro} subpockets. DeLinker demonstrated the feasibility of generative synthesis conditioned on seed fragments but also highlight the need for rationale initial fragment setup.

FeatureDock complements ChemPLAN-Net by addressing the docking and pose prediction problem from a physicochemical perspective. It employs a transformer-based architecture to model spatial distributions of ligand binding preferences as probability density envelopes over discretized protein pocket grids. By utilizing the attention maps, FeatureDock further provides visualization explanation to analyze contributing factors, facilitating model interpretation. In virtual screening, FeatureDock achieved higher early enrichment than traditional docking methods and approached the high pose accuracy. We expect that the idea of using probability density envelopes will become promising for

pocket-conditioned generative models complementary to the graph-based or sequence-based models in drug discovery.

5.2 Future work

The promising results demonstrated by ChemPLAN-Net and FeatureDock underscore several critical directions for future research and development. These directions aim to enhance data fidelity, methodological robustness, model interpretability, and generative potential, ultimately moving toward integrated, intelligent, and automated drug discovery platforms.

- **High quality dataset curation**. As deep learning models continue to grow in capacity and complexity, the quality of training data remains a primary determinant of their generalizability and reliability. Future work should focus on building automated and standardized pipelines for data crawling, preprocessing, and curation, especially for molecular systems. Current datasets, largely derived from static crystal structures and curated sequence information, do not fully capture conformational heterogeneity or dynamic interaction mechanisms. Incorporating data from molecular dynamics (MD) simulations, ensemble docking, cryo-EM maps, and experimental mutagenesis studies can enrich the training landscape, facilitating the development of dynamics-aware and functionally interpretable models. More importantly, there is a growing need to extract data from research articles and code repositories, especially for developing scientific agents.

- **Conditional generative models**. The initial success of graph-based generative models such as DeLinker suggests the feasibility of end-to-end fragment-to-compound synthesis pipelines. Extending this capability through conditional generative models, particularly those conditioned on dual fragment embeddings, holds potential to transform fragment hits into optimized drug-like molecules in an automated fashion. Moreover, the probabilistic density envelopes generated by FeatureDock offer an intriguing opportunity to serve as spatial priors for pocket-conditioned generation, aligning molecular generation directly with biophysically meaningful binding landscapes.

- **Consensus modelling**. The complexity of protein–ligand interactions and the diversity of molecular representations necessitate the use of ensemble strategies. Future work should explore consensus modelling frameworks that combine outputs from

multiple predictive or generative models to enhance reliability and reduce variance. This includes integrating structure-based and ligand-based models, as well as blending models trained on distinct data modalities (e.g., graphs, grids, and sequences). Large language model (LLM)-based agents may also play a role in orchestrating these workflows, mediating design–test across disparate model outputs.

- **Explainable AI**. Interpretability remains a central challenge in the deployment of deep learning models for drug discovery. Future directions should continue the development of explainable AI frameworks that provide mechanistic insight into model predictions.

- **Multi-modality learning**. Protein–ligand interactions are inherently multimodal, involving sequences, three-dimensional structure, dynamics, and chemical reactivity. Models that can integrate and learn from these heterogeneous data types will offer a more holistic understanding of molecular recognition. Future research should explore architectures capable of fusing various modalities.

- **LLM-agent aided computational drug discovery**.The integration of large language models (LLMs) as autonomous agents has the potential to transform the drug discovery process by coordinating tasks across chemistry, biology, and computational modeling. Unlike static models, LLM-agents can read scientific literature, query domain-specific databases, generate mechanistic hypotheses, propose implementation strategies, execute and debug code, and iteratively refine designs in a closed-loop system. Future work should emphasize strengthening their cross-modal reasoning abilities, and enabling seamless interaction with computational and experimental platforms to guide decision-making. As these agents become increasingly specialized and grounded in biomedical knowledge, they may evolve into collaborative co-pilots for researchers in the field of drug discovery.

Bibliography

[1] Zuzanna Kozicka and Nicolas Holger Thomä. Haven't got a glue: Protein surface variation for the design of molecular glue degraders. *Cell Chemical Biology*, 28(7):1032–1047, 2021.

[2] Mingyi Xue, Bojun Liu, Siqin Cao, and Xuhui Huang. Featuredock for protein-ligand docking guided by physicochemical feature-based local environment learning using transformer. *npj Drug Discovery*, 2(1):4, 2025.

[3] Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, et al. Scienceagentbench: Toward rigorous assessment of language agents for data-driven scientific discovery. *arXiv preprint arXiv:2410.05080*, 2024.

[4] Michael A Suarez Vasquez, Mingyi Xue, Jordy H Lam, Eshani C Goonetilleke, Xin Gao, and Xuhui Huang. Chemplan-net: A deep learning framework to find novel inhibitor fragments for proteins. *bioRxiv*, pages 2021–08, 2021.

[5] Tatiana F Vieira and Sérgio F Sousa. Comparing autodock and vina in ligand/decoy discrimination for virtual screening. *Applied Sciences*, 9(21):4538, 2019.

[6] Viet-Khoa Tran-Nguyen, Célien Jacquemard, and Didier Rognan. Lit-pcba: an unbiased data set for machine learning and virtual screening. *Journal of chemical information and modeling*, 60(9):4263–4273, 2020.

[7] Gregory Sliwoski, Sandeepkumar Kothiwale, Jens Meiler, and Edward W Lowe Jr. Computational methods in drug discovery. *Pharmacological reviews*, 66(1):334–395, 2014.

[8] Vincent Zoete, Aurélien Grosdidier, and Olivier Michielin. Docking, virtual high throughput screening and in silico fragment-based drug design. *Journal of cellular and molecular medicine*, 13(2):238–248, 2009.

[9] Pavel G Polishchuk, Timur I Madzhidov, and Alexandre Varnek. Estimation of the size of drug-like chemical space based on gdb-17 data. *Journal of computer-aided molecular design*, 27:675–679, 2013.

[10] Nurken Berdigaliyev and Mohamad Aljofan. An overview of drug discovery and development. *Future medicinal chemistry*, 12(10):939–947, 2020.

[11] Yeuan Ting Lee, Yi Jer Tan, and Chern Ein Oon. Molecular targeted therapy: Treating cancer with specificity. *European journal of pharmacology*, 834:188–196, 2018.

[12] Ryan G Coleman and Kim A Sharp. Protein pockets: inventory, shape, and comparison. *Journal of chemical information and modeling*, 50(4):589–603, 2010.

[13] Stéphanie Pérot, Olivier Sperandio, Maria A Miteva, Anne-Claude Camproux, and Bruno O Villoutreix. Druggable pockets and binding site centric chemical space: a paradigm shift in drug discovery. *Drug discovery today*, 15(15-16):656–667, 2010.

[14] Maria Batool, Bilal Ahmad, and Sangdun Choi. A structure-based drug discovery paradigm. *International journal of molecular sciences*, 20(11):2783, 2019.

[15] Xiao-Chen Bai, Greg McMullan, and Sjors HW Scheres. How cryo-em is revolutionizing structural biology. *Trends in biochemical sciences*, 40(1):49–57, 2015.

[16] Muhammed Tilahun Muhammed and Esin Aki-Yalcin. Homology modeling in drug discovery: Overview, current applications, and future perspectives. *Chemical biology & drug design*, 93(1):12–20, 2019.

[17] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

[18] Mihaly Varadi, Stephen Anyango, Mandar Deshpande, Sreenath Nair, Cindy Natassia, Galabina Yordanova, David Yuan, Oana Stroe, Gemma Wood, Agata Laydon, et al. Alphafold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic acids research*, 50(D1):D439–D444, 2022.

[19] Joseph C Somody, Stephen S MacKinnon, and Andreas Windemuth. Structural coverage of the proteome for pharmaceutical applications. *Drug discovery today*, 22(12):1792–1799, 2017.

[20] Vincent Le Guilloux, Peter Schmidtke, and Pierre Tuffery. Fpocket: an open source platform for ligand pocket detection. *BMC bioinformatics*, 10:1–11, 2009.

[21] José Jiménez, Stefan Doerr, Gerard Martínez-Rosell, Alexander S Rose, and Gianni De Fabritiis. Deepsite: protein-binding site predictor using 3d-convolutional neural networks. *Bioinformatics*, 33(19):3036–3042, 2017.

[22] Jacob D Durrant and J Andrew McCammon. Molecular dynamics simulations and drug discovery. *BMC biology*, 9:1–9, 2011.

[23] Vladimiras Oleinikovas, Giorgio Saladino, Benjamin P Cossins, and Francesco L Gervasio. Understanding cryptic pocket formation in protein targets by enhanced sampling simulations. *Journal of the American Chemical Society*, 138(43):14257–14263, 2016.

[24] Lei Xie, Li Xie, and Philip E Bourne. Structure-based systems biology for analyzing off-target binding. *Current opinion in structural biology*, 21(2):189–199, 2011.

[25] Chi V Dang, E Premkumar Reddy, Kevan M Shokat, and Laura Soucek. Drugging the'undruggable'cancer targets. *Nature Reviews Cancer*, 17(8):502–508, 2017.

[26] Mariell Pettersson and Craig M Crews. Proteolysis targeting chimeras (protacs)—past, present and future. *Drug Discovery Today: Technologies*, 31:15–27, 2019.

[27] Janet M Sasso, Rumiana Tenchov, DaSheng Wang, Linda S Johnson, Xinmei Wang, and Qiongqiong Angela Zhou. Molecular glues: the adhesive connecting targeted protein degradation to the clinic. *Biochemistry*, 62(3):601–623, 2022.

[28] Deborah Chirnomas, Keith R Hornberger, and Craig M Crews. Protein degraders enter the clinic—a new approach to cancer therapy. *Nature reviews Clinical oncology*, 20(4):265–278, 2023.

[29] Mike Collins, Jessica Wan, Victoria Garbitt-Amaral, Brandon Antonakos, Hsin-Jung Wu, Christina Xu, David L Lahr, Liyue Huang, Virna Schuck, Qianhe Zhou, et al. Preclinical validation of target engagement assays and investigation of mechanistic impacts of fhd-609, a clinical-stage brd9 degrader being developed for the treatment of synovial sarcoma. *strategies*, 36:936–950, 2022.

[30] Hisham Mazal and Gilad Haran. Single-molecule fret methods to study the dynamics of proteins at work. *Current opinion in biomedical engineering*, 12:8–17, 2019.

[31] Antonija Kuzmanic, Gregory R Bowman, Jordi Juarez-Jimenez, Julien Michel, and Francesco L Gervasio. Investigating cryptic binding sites by molecular dynamics simulations. *Accounts of chemical research*, 53(3):654–661, 2020.

[32] Shaoyong Lu, Mingfei Ji, Duan Ni, and Jian Zhang. Discovery of hidden allosteric sites as novel targets for allosteric drug design. *Drug discovery today*, 23(2):359–365, 2018.

[33] Brooke E Husic and Vijay S Pande. Markov state models: From an art to a science. *Journal of the American Chemical Society*, 140(7):2386–2396, 2018.

[34] Kirill A Konovalov, Ilona Christy Unarta, Siqin Cao, Eshani C Goonetilleke, and Xuhui Huang. Markov state models to study the functional dynamics of proteins in the wake of machine learning. *JACS Au*, 1(9):1330–1341, 2021.

[35] Frank Noé, Christof Schütte, Eric Vanden-Eijnden, Lothar Reich, and Thomas R Weikl. Constructing the equilibrium ensemble of folding pathways from short off-equilibrium simulations. *Proceedings of the National Academy of Sciences*, 106(45):19011–19016, 2009.

[36] Kirill A Konovalov, Cheng-Guo Wu, Yunrui Qiu, Vijaya Kumar Balakrishnan, Pankaj Singh Parihar, Michael S O'Connor, Yongna Xing, and Xuhui Huang. Disease mutations and phosphorylation alter the allosteric pathways involved in autoinhibition of protein phosphatase 2a. *The Journal of Chemical Physics*, 158(21), 2023.

[37] Siqin Cao, Yunrui Qiu, Michael L Kalin, and Xuhui Huang. Integrative generalized master equation: A method to study long-timescale biomolecular dynamics via the integrals of memory kernels. *The Journal of Chemical Physics*, 159(13), 2023.

[38] Bojun Liu, Siqin Cao, Jordan G Boysen, Mingyi Xue, and Xuhui Huang. Memory kernel minimization based neural networks for discovering slow collective variables of biomolecular dynamics. 2024.

[39] Lutz Molgedey and Heinz Georg Schuster. Separation of a mixture of independent signals using time delayed correlations. *Physical review letters*, 72(23):3634, 1994.

[40] Andreas Mardt, Luca Pasquali, Hao Wu, and Frank Noé. Vampnets for deep learning of molecular kinetics. *Nature communications*, 9(1):5, 2018.

[41] Jean-Louis Reymond and Mahendra Awale. Exploring chemical space for drug discovery using the chemical universe database. *ACS Chemical Neuroscience*, 3(9):649–657, May 2012.

[42] James P Hughes, Stephen Rees, S Barrett Kalindjian, and Karen L Philpott. Principles of early drug discovery. *British journal of pharmacology*, 162(6):1239–1249, 2011.

[43] Tiago G Fernandes, Maria Margarida Diogo, Douglas S Clark, Jonathan S Dordick, and Joaquim MS Cabral. High-throughput cellular microarray platforms: applications in drug discovery, toxicology and stem cell research. *Trends in biotechnology*, 27(6):342–349, 2009.

[44] Rohan Gupta, Devesh Srivastava, Mehar Sahu, Swati Tiwari, Rashmi K Ambasta, and Pravir Kumar. Artificial intelligence to deep learning: machine intelligence approach for drug discovery. *Molecular diversity*, 25:1315–1360, 2021.

[45] Kit-Kay Mak, Yi-Hang Wong, and Mallikarjuna Rao Pichika. Artificial intelligence in drug discovery and development. *Drug discovery and evaluation: safety and pharmacokinetic assays*, pages 1461–1498, 2024.

[46] Monica Schenone, Vlado Dančík, 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.

[47] Shenliang Wang, Tae Bo Sim, Yang-Suk Kim, and Young-Tae Chang. Tools for target identification and validation. *Current opinion in chemical biology*, 8(4):371–377, 2004.

[48] Peiling Du, Rui Fan, Nana Zhang, Chenyuan Wu, and Yingqian Zhang. Advances in integrated multi-omics analysis for drug-target identification. *Biomolecules*, 14(6):692, 2024.

[49] Theodora Katsila, Georgios A Spyroulias, George P Patrinos, and Minos-Timotheos Matsoukas. Computational approaches in target identification and drug discovery. *Computational and structural biotechnology journal*, 14:177–184, 2016.

[50] Inigo Barrio-Hernandez and Pedro Beltrao. Network analysis of genome-wide association studies for drug target prioritisation. *Current opinion in chemical biology*, 71:102206, 2022.

[51] Marina Bykova, Yuan Hou, Charis Eng, and Feixiong Cheng. Quantitative trait locus (xqtl) approaches identify risk genes and drug targets from human non-coding genomes. *Human Molecular Genetics*, 31(R1):R105–R113, 2022.

[52] Aidan MacNamara, Nikolina Nakic, Ali Amin Al Olama, Cong Guo, Karsten B Sieber, Mark R Hurle, and Alex Gutteridge. Network and pathway expansion of genetic disease associations identifies successful drug targets. *Scientific reports*, 10(1):20970, 2020.

[53] Xiangxiang Zeng, Siyi Zhu, Weiqiang Lu, Zehui Liu, Jin Huang, Yadi Zhou, Jiansong Fang, Yin Huang, Huimin Guo, Lang Li, et al. Target identification among known drugs by deep learning from heterogeneous networks. *Chemical Science*, 11(7):1775–1797, 2020.

[54] Robert A Blake. Target validation in drug discovery. *High Content Screening: A Powerful Approach to Systems Cell Biology and Drug Discovery*, pages 367–377, 2006.

[55] William G Kaelin Jr. Common pitfalls in preclinical cancer target validation. *Nature Reviews Cancer*, 17(7):441–450, 2017.

[56] Paul Carter, Leia Smith, and Maureen Ryan. Identification and validation of cell surface antigens for antibody targeting in oncology. *Endocrine-related cancer*, 11(4):659–687, 2004.

[57] Gautier Koscielny, Peter An, Denise Carvalho-Silva, Jennifer A Cham, Luca Fumis, Rippa Gasparyan, Samiul Hasan, Nikiforos Karamanis, Michael Maguire, Eliseo Papa, et al. Open targets: a platform for therapeutic target identification and validation. *Nucleic acids research*, 45(D1):D985–D994, 2017.

[58] Lorenz M. Mayr and Peter Fuerst. The future of high-throughput screening. *Journal of Biomolecular Screening*, 13(6):443–448, July 2008.

[59] Paweł Szymański, Magdalena Markowicz, and Elżbieta Mikiciuk-Olasik. Adaptation of high-throughput screening in drug discovery—toxicological screening tests. *International journal of molecular sciences*, 13(1):427–452, 2011.

[60] Robert Abel, Lingle Wang, Edward D Harder, BJ Berne, and Richard A Friesner. Advancing drug discovery through enhanced free energy calculations. *Accounts of chemical research*, 50(7):1625–1632, 2017.

[61] Lingle Wang, Yujie Wu, Yuqing Deng, Byungchan Kim, Levi Pierce, Goran Krilov, Dmitry Lupyan, Shaughnessy Robinson, Markus K Dahlgren, Jeremy Greenwood, et al. Accurate and reliable prediction of relative ligand binding potency in prospective drug discovery by way of a modern free-energy calculation protocol and force field. *Journal of the American Chemical Society*, 137(7):2695–2703, 2015.

[62] Zoe Cournia, Bryce Allen, and Woody Sherman. Relative binding free energy calculations in drug discovery: recent advances and practical considerations. *Journal of chemical information and modeling*, 57(12):2911–2937, 2017.

[63] Douglas B Kitchen, Hélène Decornez, John R Furr, and Jürgen Bajorath. Docking and scoring in virtual screening for drug discovery: methods and applications. *Nature reviews Drug discovery*, 3(11):935–949, 2004.

[64] Christoph Gorgulla, Andras Boeszoermenyi, Zi-Fu Wang, Patrick D Fischer, Paul W Coote, Krishna M Padmanabha Das, Yehor S Malets, Dmytro S Radchenko, Yurii S Moroz, David A Scott, et al. An open-source drug discovery platform enables ultra-large virtual screens. *Nature*, 580(7805):663–668, 2020.

[65] Peter Ripphausen, Britta Nisius, and Jürgen Bajorath. State-of-the-art in ligand-based virtual screening. *Drug discovery today*, 16(9-10):372–376, 2011.

[66] Andreas Bender, Jeremy L Jenkins, Qingliang Li, Sam E Adams, Edward O Cannon, and Robert C Glen. Molecular similarity: advances in methods, applications and validations in virtual screening and qsar. *Annual reports in computational chemistry*, 2:141–168, 2006.

[67] Sheng-Yong Yang. Pharmacophore modeling and applications in drug discovery: challenges and recent advances. *Drug discovery today*, 15(11-12):444–450, 2010.

[68] Muhammed Tilahun Muhammed and Esin Akı-yalcın. Pharmacophore modeling in drug discovery: methodology and current status. *Journal of the Turkish Chemical Society Section A: Chemistry*, 8(3):749–762, 2021.

[69] Peixun Liu and Wei Long. Current mathematical methods used in qsar/qspr studies. *International Journal of Molecular Sciences*, 10(5):1978–1998, 2009.

[70] Wenwen Lian, Jiansong Fang, Chao Li, Xiaocong Pang, Ai-Lin Liu, and Guan-Hua Du. Discovery of influenza a virus neuraminidase inhibitors using support vector machine and naïve bayesian models. *Molecular diversity*, 20(2):439–451, 2016.

[71] Daniel A Erlanson. Introduction to fragment-based drug discovery. *Fragment-based drug discovery and X-ray crystallography*, pages 1–32, 2012.

[72] Miles Congreve, Robin Carr, Chris Murray, and Harren Jhoti. A'rule of three'for fragment-based lead discovery? *Drug discovery today*, 8(19):876–877, 2003.

[73] Qingxin Li. Application of fragment-based drug discovery to versatile targets. *Frontiers in molecular biosciences*, 7:180, 2020.

[74] Lauro Ribeiro de Souza Neto, José Teófilo Moreira-Filho, Bruno Junior Neves, Rocío Lucía Beatriz Riveros Maidana, Ana Carolina Ramos Guimarães, Nicholas Furnham, Carolina Horta Andrade, and Floriano Paes Silva Jr. In silico strategies to support fragment-to-lead optimization in drug discovery. *Frontiers in chemistry*, 8:93, 2020.

[75] Jinhyeok Yoo, Wonkyeong Jang, and Woong-Hee Shin. From part to whole: Ai-driven progress in fragment-based drug discovery. *Current Opinion in Structural Biology*, 91:102995, 2025.

[76] Jacob D Durrant, Rommie E Amaro, and J Andrew McCammon. Autogrow: a novel algorithm for protein inhibitor design. *Chemical biology & drug design*, 73(2):168–178, 2009.

[77] Thomas E Hadfield, Fergus Imrie, Andy Merritt, Kristian Birchall, and Charlotte M Deane. Incorporating target-specific pharmacophoric information into deep generative models for fragment elaboration. *Journal of Chemical Information and Modeling*, 62(10):2280–2292, 2022.

[78] Osamu Ichihara, John Barker, Richard J Law, and Mark Whittaker. Compound design by fragment-linking. *Molecular Informatics*, 30(4):298–306, 2011.

[79] Fergus Imrie, Anthony R Bradley, Mihaela van der Schaar, and Charlotte M Deane. Deep generative models for 3d linker design. *Journal of chemical information and modeling*, 60(4):1983–1995, 2020.

[80] Stephanie Wills, Ruben Sanchez-Garcia, Tim Dudgeon, Stephen D Roughley, Andy Merritt, Roderick E Hubbard, James Davidson, Frank von Delft, and Charlotte M Deane. Fragment merging using a graph database samples different catalogue space than similarity search. *Journal of Chemical Information and Modeling*, 63(11):3423–3437, 2023.

[81] Elizabeth V Bedwell, William J McCarthy, Anthony G Coyne, and Chris Abell. Development of potent inhibitors by fragment-linking strategies. *Chemical Biology & Drug Design*, 100(4):469–486, 2022.

[82] Tobias Wunberg, Martin Hendrix, Alexander Hillisch, Mario Lobell, Heinrich Meier, Carsten Schmeck, Hanno Wild, and Berthold Hinzen. Improving the hit-to-lead process: data-driven assessment of drug-like and lead-like screening hits. *Drug discovery today*, 11(3-4):175–180, 2006.

[83] György M Keserű and Gergely M Makara. Hit discovery and hit-to-lead approaches. *Drug discovery today*, 11(15-16):741–748, 2006.

[84] Sarah R Langdon, Peter Ertl, and Nathan Brown. Bioisosteric replacement and scaffold hopping in lead generation and optimization. *Molecular informatics*, 29(5):366–385, 2010.

[85] Alleyn T Plowright, Craig Johnstone, Jan Kihlberg, Jonas Pettersson, Graeme Robb, and Richard A Thompson. Hypothesis driven drug design: improving quality and effectiveness of the design-make-test-analyse cycle. *Drug discovery today*, 17(1-2):56–62, 2012.

[86] Gian Marco Ghiandoni, Emma Evertsson, David J Riley, Christian Tyrchan, and Prakash Chandra Rathi. Augmenting dmta using predictive ai modelling at astrazeneca. *Drug discovery today*, 29(4):103945, 2024.

[87] Sunghoon Joo, Min Soo Kim, Jaeho Yang, and Jeahyun Park. Generative model for proposing drug candidates satisfying anticancer properties using a conditional variational autoencoder. *ACS omega*, 5(30):18642–18650, 2020.

[88] Mariya Popova, Olexandr Isayev, and Alexander Tropsha. Deep reinforcement learning for de novo drug design. *Science advances*, 4(7):eaap7885, 2018.

[89] Alex Zhavoronkov, Yan A Ivanenkov, Alex Aliper, Mark S Veselov, Vladimir A Aladinskiy, Anastasiya V Aladinskaya, Victor A Terentiev, Daniil A Polykovskiy, Maksim D Kuznetsov, Arip Asadulaev, et al. Deep learning enables rapid identification of potent ddr1 kinase inhibitors. *Nature biotechnology*, 37(9):1038–1040, 2019.

[90] Yanli Wang, Jewen Xiao, Tugba O Suzek, Jian Zhang, Jiyao Wang, and Stephen H Bryant. Pubchem: a public information system for analyzing bioactivities of small molecules. *Nucleic acids research*, 37(suppl_2):W623–W633, 2009.

[91] Sunghwan Kim, Paul A Thiessen, Evan E Bolton, Jie Chen, Gang Fu, Asta Gindulyte, Lianyi Han, Jane He, Siqian He, Benjamin A Shoemaker, et al. Pubchem substance and compound databases. *Nucleic acids research*, 44(D1):D1202–D1213, 2016.

[92] Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, et al. Pubchem 2019 update: improved access to chemical data. *Nucleic acids research*, 47(D1):D1102–D1109, 2019.

[93] Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, et al. Pubchem 2023 update. *Nucleic acids research*, 51(D1):D1373–D1380, 2023.

[94] 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.

[95] John J Irwin, Teague Sterling, Michael M Mysinger, Erin 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.

[96] John J Irwin, Khanh G Tang, Jennifer Young, Chinzorig Dandarchuluun, Benjamin R Wong, Munkhzul Khurelbaatar, Yurii S Moroz, John Mayfield, and Roger A Sayle. Zinc20—a free ultralarge-scale chemical database for ligand discovery. *Journal of chemical information and modeling*, 60(12):6065–6073, 2020.

[97] Benjamin I Tingle, Khanh G Tang, Mar Castanon, John J Gutierrez, Munkhzul Khurelbaatar, Chinzorig Dandarchuluun, Yurii S Moroz, and John J Irwin. Zinc-22- a free multi-billion-scale database of tangible compounds for ligand discovery. *Journal of chemical information and modeling*, 63(4):1166–1176, 2023.

[98] Anna Gaulton, Louisa J Bellis, A Patricia Bento, Jon Chambers, Mark Davies, Anne Hersey, Yvonne Light, Shaun McGlinchey, David Michalovich, Bissan Al-Lazikani, et al. Chembl: a large-scale bioactivity database for drug discovery. *Nucleic acids research*, 40(D1):D1100–D1107, 2012.

[99] A Patrícia Bento, Anna Gaulton, Anne Hersey, Louisa J Bellis, Jon Chambers, Mark Davies, Felix A Krüger, Yvonne Light, Lora Mak, Shaun McGlinchey, et al. The chembl bioactivity database: an update. *Nucleic acids research*, 42(D1):D1083–D1090, 2014.

[100] Anna Gaulton, Anne Hersey, Michał Nowotka, A Patricia Bento, Jon Chambers, David Mendez, Prudence Mutowo, Francis Atkinson, Louisa J Bellis, Elena Cibrián-Uhalte, et al. The chembl database in 2017. *Nucleic acids research*, 45(D1):D945–D954, 2017.

[101] Barbara Zdrazil, Eloy Felix, Fiona Hunter, Emma J Manners, James Blackshaw, Sybilla Corbett, Marleen de Veij, Harris Ioannidis, David Mendez Lopez, Juan F Mosquera, et al. The chembl database in 2023: a drug discovery platform spanning multiple bioactivity data types and time periods. *Nucleic acids research*, 52(D1):D1180–D1192, 2024.

[102] Daniel C Elton, Zois Boukouvalas, Mark D Fuge, and Peter W Chung. Deep learning for molecular design—a review of the state of the art. *Molecular Systems Design & Engineering*, 4(4):828–849, 2019.

[103] Douglas B Kell, Soumitra Samanta, and Neil Swainston. Deep learning and generative methods in cheminformatics and chemical biology: navigating small molecule space intelligently. *Biochemical Journal*, 477(23):4559–4580, 2020.

[104] Wenhao Gao, Sai Pooja Mahajan, Jeremias Sulam, and Jeffrey J Gray. Deep learning in protein structural modeling and design. *Patterns*, 1(9), 2020.

[105] Alexander Kroll, Sahasra Ranjan, Martin KM Engqvist, and Martin J Lercher. A general model to predict small molecule substrates of enzymes based on machine and deep learning. *Nature communications*, 14(1):2787, 2023.

[106] Hamed Khakzad, Ilia Igashov, Arne Schneuing, Casper Goverde, Michael Bronstein, and Bruno Correia. A new age in protein design empowered by deep learning. *Cell Systems*, 14(11):925–939, 2023.

[107] Bojun Liu, Mingyi Xue, Yunrui Qiu, Kirill A Konovalov, Michael S O'Connor, and Xuhui Huang. Graphvampnets for uncovering slow collective variables of self-assembly dynamics. *The Journal of Chemical Physics*, 159(9), 2023.

[108] Yunrui Qiu, Michael S O'Connor, Mingyi Xue, Bojun Liu, and Xuhui Huang. An efficient path classification algorithm based on variational autoencoder to identify metastable path channels for complex conformational changes. *Journal of chemical theory and computation*, 19(14):4728–4742, 2023.

[109] Christopher A Lipinski, Franco Lombardo, Beryl W Dominy, and Paul J Feeney. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Advanced drug delivery reviews*, 64:4–17, 2012.

[110] Chun Wei Yap. Padel-descriptor: An open source software to calculate molecular descriptors and fingerprints. *Journal of computational chemistry*, 32(7):1466–1474, 2011.

[111] Christoph Steinbeck, Yongquan Han, Stefan Kuhn, Oliver Horlacher, Edgar Luttmann, and Egon Willighagen. The chemistry development kit (cdk): An open-source java library for chemo-and bioinformatics. *Journal of chemical information and computer sciences*, 43(2):493–500, 2003.

[112] Egon L Willighagen, John W Mayfield, Jonathan Alvarsson, Arvid Berg, Lars Carlsson, Nina Jeliazkova, Stefan Kuhn, Tomáš Pluskal, Miquel Rojas-Chertó, Ola Spjuth, et al. The chemistry development kit (cdk) v2. 0: atom typing, depiction, molecular formulas, and substructure searching. *Journal of cheminformatics*, 9:1–19, 2017.

[113] 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.

[114] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50(5):742–754, 2010.

[115] Adrià Cereto-Massagué, María José Ojeda, Cristina Valls, Miquel Mulero, Santiago Garcia-Vallvé, and Gerard Pujadas. Molecular fingerprint similarity search in virtual screening. *Methods*, 71:58–63, 2015.

[116] Jeremy Desaphy, Eric Raimbaud, Pierre Ducrot, and Didier Rognan. Encoding protein–ligand interaction patterns in fingerprints and graphs. *Journal of chemical information and modeling*, 53(3):623–637, 2013.

[117] Cédric Bouysset and Sébastien Fiorucci. Prolif: a library to encode molecular interactions as fingerprints. *Journal of cheminformatics*, 13(1):72, 2021.

[118] Aleksey Porollo and Jarosław Meller. Prediction-based fingerprints of protein–protein interactions. *Proteins: Structure, Function, and Bioinformatics*, 66(3):630–645, 2007.

[119] Francesca Spyrakis, Paolo Benedetti, Sergio Decherchi, Walter Rocchia, Andrea Cavalli, Stefano Alcaro, Francesco Ortuso, Massimo Baroni, and Gabriele Cruciani. A pipeline to enhance ligand virtual screening: integrating molecular dynamics and fingerprints for ligand and proteins. *Journal of Chemical Information and Modeling*, 55(10):2256–2274, 2015.

[120] Sereina Riniker. Molecular dynamics fingerprints (mdfp): machine learning from md data to predict free-energy differences. *Journal of chemical information and modeling*, 57(4):726–741, 2017.

[121] Chanin Nantasenamat, Chartchalerm Isarankura-Na-Ayudhya, Thanakorn Naenna, and Virapong Prachayasittikul. A practical overview of quantitative structure-activity relationship. *EXCLI J*, 8(7):74–88, 2009.

[122] Pavel Polishchuk. Interpretation of quantitative structure–activity relationship models: past, present, and future. *Journal of Chemical Information and Modeling*, 57(11):2618–2639, 2017.

[123] Keir Adams, Lagnajit Pattanaik, and Connor W Coley. Learning 3d representations of molecular chirality with invariance to bond rotations. *arXiv preprint arXiv:2110.04383*, 2021.

[124] Alexander Kensert, Jonathan Alvarsson, Ulf Norinder, and Ola Spjuth. Evaluating parameters for ligand-based modeling with random forest on sparse data sets. *Journal of cheminformatics*, 10(1):49, 2018.

[125] Patrick Hop, Brandon Allgood, and Jessen Yu. Geometric deep learning autonomously learns chemical features that outperform those engineered by domain experts. *Molecular pharmaceutics*, 15(10):4371–4377, 2018.

[126] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.

[127] David Weininger, Arthur Weininger, and Joseph L Weininger. Smiles. 2. algorithm for generation of unique smiles notation. *Journal of chemical information and computer sciences*, 29(2):97–101, 1989.

[128] Greg Landrum, Paolo Tosco, Brian Kelley, Ricardo Rodriguez, David Cosgrove, Riccardo Vianello, sriniker, Peter Gedeck, Gareth Jones, Eisuke Kawashima, NadineSchneider, Dan Nealschneider, Andrew Dalke, Matt Swain, Brian Cole, Samo Turk, Aleksandr Savelev, tadhurst-cdd, Alain Vaucher, Maciej Wójcikowski, Ichiru Take, Rachel Walker, Vincent F Scalfani, Hussein Faara, Kazuya Ujihara, Daniel Probst, Juuso Lehtivarjo, Guillaume Godin, Axel Pahl, and Niels Maeder. rdkit/rdkit: 2025_03_2 (q1 2025) release, 2025.

[129] Harry L Morgan. The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service. *Journal of chemical documentation*, 5(2):107–113, 1965.

[130] Nadine Schneider, Roger A Sayle, and Gregory A Landrum. Get your atoms in order - an open-source implementation of a novel and robust molecular canonicalization algorithm. *Journal of chemical information and modeling*, 55(10):2111–2120, 2015.

[131] Noel M O'Boyle. Towards a universal smiles representation-a standard method to generate canonical smiles based on the inchi. *Journal of cheminformatics*, 4:1–14, 2012.

[132] Chunyan Li, Jihua Feng, Shihu Liu, and Junfeng Yao. A novel molecular representation learning for molecular property prediction with a multiple smiles-based augmentation. *Computational Intelligence and Neuroscience*, 2022(1):8464452, 2022.

[133] Garrett B Goh, Nathan O Hodas, Charles Siegel, and Abhinav Vishnu. Smiles2vec: An interpretable general-purpose deep neural network for predicting chemical properties. *arXiv preprint arXiv:1712.02034*, 2017.

[134] Xiaochu Tong, Xiaohong Liu, Xiaoqin Tan, Xutong Li, Jiaxin Jiang, Zhaoping Xiong, Tingyang Xu, Hualiang Jiang, Nan Qiao, and Mingyue Zheng. Generative models for de novo drug design. *Journal of Medicinal Chemistry*, 64(19):14011–14027, 2021.

[135] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

[136] Thomas Blaschke, Marcus Olivecrona, Ola Engkvist, Jürgen Bajorath, and Hongming Chen. Application of generative autoencoder in de novo molecular design. *Molecular informatics*, 37(1-2):1700123, 2018.

[137] Maria Korshunova, Niles Huang, Stephen Capuzzi, Dmytro S Radchenko, Olena Savych, Yuriy S Moroz, Carrow I Wells, Timothy M Willson, Alexander Tropsha, and Olexandr Isayev. Generative and reinforcement learning approaches for the automated de novo design of bioactive compounds. *Communications Chemistry*, 5(1):129, 2022.

[138] Noel O'Boyle and Andrew Dalke. Deepsmiles: an adaptation of smiles for use in machine-learning of chemical structures. 2018.

[139] Mario Krenn, Florian Häse, AkshatKumar Nigam, Pascal Friederich, and Alan Aspuru-Guzik. Self-referencing embedded strings (selfies): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, 1(4):045024, 2020.

[140] Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, and Hongming Chen. Molecular de-novo design through deep reinforcement learning. *Journal of cheminformatics*, 9:1–14, 2017.

[141] Stephen R Heller, Alan McNaught, Igor Pletnev, Stephen Stein, and Dmitrii Tchekhovskoi. Inchi, the iupac international chemical identifier. *Journal of cheminformatics*, 7:1–34, 2015.

[142] InChI Homepage - InChI Trust — inchi-trust.org. `http://www.inchi-trust.org/`. [Accessed 28-05-2025].

[143] Philippe Schwaller, Daniel Probst, Alain C Vaucher, Vishnu H Nair, David Kreutter, Teodoro Laino, and Jean-Louis Reymond. Mapping the space of chemical reactions using attention-based neural networks. *Nature machine intelligence*, 3(2):144–152, 2021.

[144] Philippe Schwaller, Teodoro Laino, Théophile Gaudin, Peter Bolgar, Christopher A Hunter, Costas Bekas, and Alpha A Lee. Molecular transformer: a model for uncertainty-calibrated chemical reaction prediction. *ACS central science*, 5(9):1572–1583, 2019.

[145] 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.

[146] Philippe Schwaller, Theophile Gaudin, David Lanyi, Costas Bekas, and Teodoro Laino. "found in translation": predicting outcomes of complex organic chemistry reactions using neural sequence-to-sequence models. *Chemical science*, 9(28):6091–6098, 2018.

[147] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*, 2018.

[148] Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. Chemberta: large-scale self-supervised pretraining for molecular property prediction. *arXiv preprint arXiv:2010.09885*, 2020.

[149] Tzyy-Shyang Lin, Connor W Coley, Hidenobu Mochigase, Haley K Beech, Wencong Wang, Zi Wang, Eliot Woods, Stephen L Craig, Jeremiah A Johnson, Julia A Kalow, et al. Bigsmiles: a structurally-based line notation for describing macromolecules. *ACS central science*, 5(9):1523–1531, 2019.

[150] Roshan Rao, Joshua Meier, Tom Sercu, Sergey Ovchinnikov, and Alexander Rives. Transformer protein language models are unsupervised structure learners. *Biorxiv*, pages 2020–12, 2020.

[151] Thomas Hayes, Roshan Rao, Halil Akin, Nicholas J Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q Tran, Jonathan Deaton, Marius Wiggert, et al. Simulating 500 million years of evolution with a language model. *Science*, page eads0018, 2025.

[152] 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.

[153] Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, 2024.

[154] Tianyu Wu, Yang Tang, Qiyu Sun, and Luolin Xiong. Molecular joint representation learning via multi-modal information of smiles and graphs. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 20(5):3044–3055, 2023.

[155] Hannah K Wayment-Steele, Adedolapo Ojoawo, Renee Otten, Julia M Apitz, Warintra Pitsawong, Marc Hömberger, Sergey Ovchinnikov, Lucy Colwell, and Dorothee Kern. Predicting multiple conformations via sequence clustering and alphafold2. *Nature*, 625(7996):832–839, 2024.

[156] Davide Sala, Felix Engelberger, Hassane S Mchaourab, and Jens Meiler. Modeling conformational states of proteins with alphafold. *Current Opinion in Structural Biology*, 81:102645, 2023.

[157] Patrick Bryant, Gabriele Pozzati, Wensi Zhu, Aditi Shenoy, Petras Kundrotas, and Arne Elofsson. Predicting the structure of large protein complexes using alphafold and monte carlo tree search. *Nature communications*, 13(1):6028, 2022.

[158] Amy X Lu, Wilson Yan, Sarah A Robinson, Simon Kelow, Kevin K Yang, Vladimir Gligorijevic, Kyunghyun Cho, Richard Bonneau, Pieter Abbeel, and Nathan C Frey. All-atom protein generation with latent diffusion. In *ICLR 2025 Workshop on Generative and Experimental Perspectives for Biomolecular Design*.

[159] Helen M Berman, Tammy Battistuz, Talapady N Bhat, Wolfgang F Bluhm, Philip E Bourne, Kyle Burkhardt, Zukang Feng, Gary L Gilliland, Lisa Iype, Shri Jain, et al. The protein data bank. *Biological Crystallography*, 58(6):899–907, 2002.

[160] Lagnajit Pattanaik and Connor W Coley. Molecular representation: going long on fingerprints. *Chem*, 6(6):1204–1207, 2020.

[161] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. *Advances in neural information processing systems*, 28, 2015.

[162] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

[163] Shuangli Li, Jingbo Zhou, Tong Xu, Liang Huang, Fan Wang, Haoyi Xiong, Weili Huang, Dejing Dou, and Hui Xiong. Structure-aware interactive graph neural networks for the prediction of protein-ligand binding affinity. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 975–985, 2021.

[164] Mahdi Ghorbani, Samarjeet Prasad, Jeffery B Klauda, and Bernard R Brooks. Graphvampnet, using graph neural networks and variational approach to markov processes for dynamical modeling of biomolecules. *The Journal of Chemical Physics*, 156(18), 2022.

[165] Martin Simonovsky and Nikos Komodakis. Graphvae: Towards generation of small graphs using variational autoencoders. In *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part I 27*, pages 412–422. Springer, 2018.

[166] Nicola De Cao and Thomas Kipf. Molgan: An implicit generative model for small molecular graphs. *arXiv preprint arXiv:1805.11973*, 2018.

[167] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Hierarchical generation of molecular graphs using structural motifs. In *International conference on machine learning*, pages 4839–4848. PMLR, 2020.

[168] Ziqi Gao, Chenran Jiang, Jiawen Zhang, Xiaosen Jiang, Lanqing Li, Peilin Zhao, Huanming Yang, Yong Huang, and Jia Li. Hierarchical graph learning for protein–protein interaction. *Nature Communications*, 14(1):1093, 2023.

[169] Izhar Wallach, Michael Dzamba, and Abraham Heifets. Atomnet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery. *arXiv preprint arXiv:1510.02855*, 2015.

[170] José Jiménez, Miha Skalic, Gerard Martinez-Rosell, and Gianni De Fabritiis. K deep: protein–ligand absolute binding affinity prediction via 3d-convolutional neural networks. *Journal of chemical information and modeling*, 58(2):287–296, 2018.

[171] Jordy Homing Lam, Yu Li, Lizhe Zhu, Ramzan Umarov, Hanlun Jiang, Amélie Héliou, Fu Kit Sheong, Tianyun Liu, Yongkang Long, Yunfei Li, et al. A deep learning framework to predict binding preference of rna constituents on protein surface. *Nature communications*, 10(1):4941, 2019.

[172] Soroush Ahmadi, Mohammad Amin Ghanavati, and Sohrab Rohani. Machine learning-guided prediction of cocrystals using point cloud-based molecular representation. *Chemistry of Materials*, 36(3):1153–1161, 2024.

[173] Prasoon Kumar Vinodkumar, Dogus Karabulut, Egils Avots, Cagri Ozcinar, and Gholamreza Anbarjafari. A survey on deep learning based segmentation, detection and classification for 3d point clouds. *Entropy*, 25(4):635, 2023.

[174] Emiel Hoogeboom, Vıctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *International conference on machine learning*, pages 8867–8887. PMLR, 2022.

[175] Andrea Mauri, Viviana Consonni, Manuela Pavan, Roberto Todeschini, et al. Dragon software: An easy approach to molecular descriptor calculations. *Match*, 56(2):237–248, 2006.

[176] Hirotomo Moriwaki, Yu-Shi Tian, Norihito Kawashita, and Tatsuya Takagi. Mordred: a molecular descriptor calculator. *Journal of cheminformatics*, 10(1):4, 2018.

[177] Kaggle Blog – Medium — blog.kaggle.com. `http://blog.kaggle.com/2012/11/01/deep-learning-how-i-did-it-merck-1st-place-interview/`. [Accessed 29-05-2025].

[178] Bharath Ramsundar, Bowen Liu, Zhenqin Wu, Andreas Verras, Matthew Tudor, Robert P Sheridan, and Vijay Pande. Is multitask deep learning practical for pharma? *Journal of chemical information and modeling*, 57(8):2068–2076, 2017.

[179] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? *Advances in neural information processing systems*, 31, 2018.

[180] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[181] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[182] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[183] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[184] Shifa Zhong, Jiajie Hu, Xiong Yu, and Huichun Zhang. Molecular image-convolutional neural network (cnn) assisted qsar models for predicting contaminant reactivity toward oh radicals: Transfer learning, data augmentation and model interpretation. *Chemical Engineering Journal*, 408:127998, 2021.

[185] Maya Hirohara, Yutaka Saito, Yuki Koda, Kengo Sato, and Yasubumi Sakakibara. Convolutional neural network based on smiles representation of compounds for detecting chemical motif. *BMC bioinformatics*, 19:83–94, 2018.

[186] Anna M Puszkarska, Bruck Taddese, Jefferson Revell, Graeme Davies, Joss Field, David C Hornigold, Andrew Buchanan, Tristan J Vaughan, and Lucy J Colwell. Machine learning designs new gcgr/glp-1r dual agonists with enhanced biological potency. *Nature Chemistry*, 16(9):1436–1444, 2024.

[187] Joseph Gomes, Bharath Ramsundar, Evan N Feinberg, and Vijay S Pande. Atomic convolutional networks for predicting protein-ligand binding affinity. *arXiv preprint arXiv:1703.10603*, 2017.

[188] Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[189] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[190] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[191] Josep Arús-Pous, Simon Viet Johansson, Oleksii Prykhodko, Esben Jannik Bjerrum, Christian Tyrchan, Jean-Louis Reymond, Hongming Chen, and Ola Engkvist. Randomized smiles strings improve the quality of molecular generative models. *Journal of cheminformatics*, 11:1–13, 2019.

[192] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28, 2015.

[193] Bowen Liu, Bharath Ramsundar, Prasad Kawthekar, Jade Shi, Joseph Gomes, Quang Luu Nguyen, Stephen Ho, Jack Sloane, Paul Wender, and Vijay Pande. Retrosynthetic reaction prediction using neural sequence-to-sequence models. *ACS central science*, 3(10):1103–1113, 2017.

[194] 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.

[195] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[196] Kristof T Schütt, Huziel E Sauceda, P-J Kindermans, Alexandre Tkatchenko, and K-R Müller. Schnet–a deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24), 2018.

[197] Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3438–3445, 2020.

[198] Justin Airas and Bin Zhang. Scaling graph neural networks to large proteins. *Journal of chemical theory and computation*, 21(4):2055–2066, 2025.

[199] Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E Smidt, and Boris Kozinsky. E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature communications*, 13(1):2453, 2022.

[200] Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR, 2021.

[201] Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. Se (3)-transformers: 3d roto-translation equivariant attention networks. *Advances in neural information processing systems*, 33:1970–1981, 2020.

[202] 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*, 30, 2017.

[203] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[204] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[205] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297, 2020.

[206] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pages 4651–4664. PMLR, 2021.

[207] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[208] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[209] 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.

[210] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(10):7112–7127, 2021.

[211] 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. *Journal of machine learning research*, 21(140):1–67, 2020.

[212] Qizhi Pei, Wei Zhang, Jinhua Zhu, Kehan Wu, Kaiyuan Gao, Lijun Wu, Yingce Xia, and Rui Yan. Biot5: Enriching cross-modal integration in biology with chemical knowledge and natural language associations. *arXiv preprint arXiv:2310.07276*, 2023.

[213] Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The ai scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.

[214] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[215] Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, et al. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*, 2023.

[216] Xiangru Tang, Bill Qian, Rick Gao, Jiakang Chen, Xinyun Chen, and Mark B Gerstein. Biocoder: a benchmark for bioinformatics code generation with large language models. *Bioinformatics*, 40(Supplement_1):i266–i276, 2024.

[217] Nikita Mehandru, Amanda K Hall, Olesya Melnichenko, Yulia Dubinina, Daniel Tsirulnikov, David Bamman, Ahmed Alaa, Scott Saponas, and Venkat S Malladi. Bioagents: Democratizing bioinformatics analysis with multi-agent systems. *arXiv preprint arXiv:2501.06314*, 2025.

[218] Shuo Ren, Pu Jian, Zhenjiang Ren, Chunlin Leng, Can Xie, and Jiajun Zhang. Towards scientific intelligence: A survey of llm-based scientific agents. *arXiv preprint arXiv:2503.24047*, 2025.

[219] Diederik P Kingma, Max Welling, et al. Auto-encoding variational bayes, 2013.

[220] Diederik P Kingma, Max Welling, et al. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.

[221] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning internal representations by error propagation, 1985.

[222] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[223] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021.

[224] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[225] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[226] Gabriele Corso, Hannes Stärk, Bowen Jing, Regina Barzilay, and Tommi Jaakkola. Diffdock: Diffusion steps, twists, and turns for molecular docking. *arXiv preprint arXiv:2210.01776*, 2022.

[227] Grace W Tang and Russ B Altman. Knowledge-based fragment binding prediction. *PLoS computational biology*, 10(4):e1003589, 2014.

[228] Ho Ming Lam. *Exploring the fragment subspace of human CDK2 with enhanced sampling techniques and knowledge-based predictions*. PhD thesis, 2017.

[229] Christine Zardecki, Shuchismita Dutta, David S Goodsell, Robert Lowe, Maria Voigt, and Stephen K Burley. Pdb-101: Educational resources supporting molecular explorations through biology and medicine. *Protein Science*, 31(1):129–140, 2022.

[230] Stephen M Omohundro. Five balltree construction algorithms. 1989.

[231] Greg Landrum, Paolo Tosco, Brian Kelley, Ricardo Rodriguez, David Cosgrove, Riccardo Vianello, sriniker, Peter Gedeck, Gareth Jones, Eisuke Kawashima, NadineSchneider, Dan Nealschneider, Andrew Dalke, Matt Swain, Brian Cole, Samo Turk, Aleksandr Savelev, tadhurst cdd, Alain Vaucher, Maciej Wójcikowski, Ichiru Take, Rachel Walker, Vincent F. Scalfani, Hussein Faara, Kazuya Ujihara, Daniel Probst, Juuso Lehtivarjo, guillaume godin, Axel Pahl, and Niels Maeder. rdkit/rdkit: 2025_03_2 (q1 2025) release, April 2025.

[232] Peter JA Cock, Tiago Antao, Jeffrey T Chang, Brad A Chapman, Cymon J Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, et al. Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422, 2009.

[233] Syed Asad Rahman, Matthew Bashton, Gemma L Holliday, Rainer Schrader, and Janet M Thornton. Small molecule subgraph detector (smsd) toolkit. *Journal of cheminformatics*, 1:1–13, 2009.

[234] Céline M Labbé, Mélaine A Kuenemann, Barbara Zarzycka, Gert Vriend, Gerry AF Nicolaes, David Lagorce, Maria A Miteva, Bruno O Villoutreix, and Olivier Sperandio. ippi-db: an online database of modulators of protein–protein interactions. *Nucleic acids research*, 44(D1):D542–D547, 2016.

[235] Inbal Halperin, Dariya S Glazer, Shirley Wu, and Russ B Altman. The feature framework for protein function annotation: modeling new functions, improving performance, and extending to novel applications. *BMC genomics*, 9(Suppl 2):S2, 2008.

[236] Heba T Abdel-Mohsen, Manal M Anwar, Nesreen S Ahmed, Somaia S Abd El-Karim, and Sameh H Abdelwahed. Recent advances in structural optimization of quinazoline-based protein kinase inhibitors for cancer therapy (2021–present). *Molecules*, 29(4):875, 2024.

[237] Haiying Lu, Qiaodan Zhou, Jun He, Zhongliang Jiang, Cheng Peng, Rongsheng Tong, and Jianyou Shi. Recent advances in the development of protein–protein interactions modulators: mechanisms and clinical trials. *Signal transduction and targeted therapy*, 5(1):213, 2020.

[238] Shao Jinsong, Jia Qifeng, Chen Xing, Yajie Hao, and Li Wang. Molecular fragmentation as a crucial step in the ai-based drug development pathway. *Communications Chemistry*, 7(1):20, 2024.

[239] 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.

[240] Yuyao Yang, Shuangjia Zheng, Shimin Su, Chao Zhao, Jun Xu, and Hongming Chen. Syntalinker: automatic fragment linking with deep conditional transformer neural networks. *Chemical science*, 11(31):8312–8322, 2020.

[241] Andrew G Leach, Huw D Jones, David A Cosgrove, Peter W Kenny, Linette Ruston, Philip MacFaul, J Matthew Wood, Nicola Colclough, and Brian Law. Matched molecular pairs as a guide in the optimization of pharmaceutical properties; a study of aqueous solubility, plasma protein binding and oral exposure. *Journal of medicinal chemistry*, 49(23):6672–6682, 2006.

[242] Ed Griffen, Andrew G Leach, Graeme R Robb, and Daniel J Warner. Matched molecular pairs as a medicinal chemistry tool: miniperspective. *Journal of medicinal chemistry*, 54(22):7739–7750, 2011.

[243] Alexander G Dossetter, Edward J Griffen, and Andrew G Leach. Matched molecular pair analysis in drug discovery. *Drug Discovery Today*, 18(15-16):724–731, 2013.

[244] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[245] Michael Alexander Suarez Vasquez. *Deep learning methods to find potential inhibitor fragments for proteins*. PhD thesis, 2021.

[246] Brett L Roberts, Zhi-Xiong Ma, Ang Gao, Eric D Leisten, Dan Yin, Wei Xu, and Weiping Tang. Two-stage strategy for development of proteolysis targeting chimeras and its application for estrogen receptor degraders. *ACS chemical biology*, 15(6):1487–1496, 2020.

[247] Le Guo, Jin Liu, Xueqing Nie, Taobo Wang, Zhi-xiong Ma, Dan Yin, and Weiping Tang. Development of selective fgfr1 degraders using a rapid synthesis of proteolysis targeting chimera (rapid-tac) platform. *Bioorganic & Medicinal Chemistry Letters*, 75:128982, 2022.

[248] Ni Liu and Zhibin Xu. Using ledock as a docking tool for computational drug design. In *IOP Conference Series: Earth and Environmental Science*, volume 218, page 012143. IOP Publishing, 2019.

[249] Jie Liu and Renxiao Wang. Classification of current scoring functions. *Journal of chemical information and modeling*, 55(3):475–482, 2015.

[250] Gregory L Warren, C Webster Andrews, Anna-Maria Capelli, Brian Clarke, Judith LaLonde, Millard H Lambert, Mika Lindvall, Neysa Nevins, Simon F Semus, Stefan Senger, et al. A critical assessment of docking programs and scoring functions. *Journal of medicinal chemistry*, 49(20):5912–5931, 2006.

[251] Inbal Halperin, Buyong Ma, Haim Wolfson, and Ruth Nussinov. Principles of docking: An overview of search algorithms and a guide to scoring functions. *Proteins: Structure, Function, and Bioinformatics*, 47(4):409–443, 2002.

[252] Irwin D Kuntz, Jeffrey M Blaney, Stuart J Oatley, Robert Langridge, and Thomas E Ferrin. A geometric approach to macromolecule-ligand interactions. *Journal of molecular biology*, 161(2):269–288, 1982.

[253] Todd JA Ewing, Shingo Makino, A Geoffrey Skillman, and Irwin D Kuntz. Dock 4.0: search strategies for automated molecular docking of flexible molecule databases. *Journal of computer-aided molecular design*, 15:411–428, 2001.

[254] William J Allen, Trent E Balius, Sudipto Mukherjee, Scott R Brozell, Demetri T Moustakas, P Therese Lang, David A Case, Irwin D Kuntz, and Robert C Rizzo. Dock 6: Impact of new features and current docking performance. *Journal of computational chemistry*, 36(15):1132–1156, 2015.

[255] David S Goodsell, Garrett M Morris, and Arthur J Olson. Automated docking of flexible ligands: applications of autodock. *Journal of molecular recognition*, 9(1):1–5, 1996.

[256] Kaushik Raha and Kenneth M Merz. A quantum mechanics-based scoring function: study of zinc ion-mediated ligand binding. *Journal of the American Chemical Society*, 126(4):1020–1021, 2004.

[257] Oleg Khoruzhii, Alexander G Donchev, Nikolay Galkin, Alexei Illarionov, Mikhail Olevanov, Vladimir Ozrin, Cary Queen, and Vladimir Tarasov. Application of a polarizable force field to calculations of relative protein–ligand binding affinities. *Proceedings of the National Academy of Sciences*, 105(30):10378–10383, 2008.

[258] Isabella A Guedes, Felipe SS Pereira, and Laurent E Dardenne. Empirical scoring functions for structure-based virtual screening: applications, critical aspects, and challenges. *Frontiers in pharmacology*, 9:1089, 2018.

[259] Lukas P Pason and Christoph A Sotriffer. Empirical scoring functions for affinity prediction of protein-ligand complexes. *Molecular informatics*, 35(11-12):541–548, 2016.

[260] O Trott and A Olson. Software news and update autodock vina: Improving the speed and accuracy of docking with a new scoring function. *Effic. Optim. Multithreading*, 31:455–461, 2009.

[261] David Ryan Koes, Matthew P Baumgartner, and Carlos J Camacho. Lessons learned in empirical scoring with smina from the csar 2011 benchmarking exercise. *Journal of chemical information and modeling*, 53(8):1893–1904, 2013.

[262] Mathew R Koebel, Grant Schmadeke, Richard G Posner, and Suman Sirimulla. Autodock vinaxb: implementation of xbsf, new empirical halogen bond scoring function, into autodock vina. *Journal of cheminformatics*, 8:1–8, 2016.

[263] Anita K Nivedha, David F Thieker, Spandana Makeneni, Huimin Hu, and Robert J Woods. Vina-carb: improving glycosidic angles during carbohydrate docking. *Journal of chemical theory and computation*, 12(2):892–901, 2016.

[264] Rodrigo Quiroga and Marcos A Villarreal. Vinardo: A scoring function based on autodock vina improves scoring, docking, and virtual screening. *PloS one*, 11(5):e0155183, 2016.

[265] Andrew T McNutt, Paul Francoeur, Rishal Aggarwal, Tomohide Masuda, Rocco Meli, Matthew Ragoza, Jocelyn Sunseri, and David Ryan Koes. Gnina 1.0: molecular docking with deep learning. *Journal of cheminformatics*, 13(1):43, 2021.

[266] Jerome Eberhardt, Diogo Santos-Martins, Andreas F Tillack, and Stefano Forli. Autodock vina 1.2. 0: New docking methods, expanded force field, and python bindings. *Journal of chemical information and modeling*, 61(8):3891–3898, 2021.

[267] Dariusz Plewczynski, Michał Łaźniewski, Rafał Augustyniak, and Krzysztof Ginalski. Can we trust docking results? evaluation of seven commonly used programs on pdbbind database. *Journal of computational chemistry*, 32(4):742–755, 2011.

[268] Minyi Su, Qifan Yang, Yu Du, Guoqin Feng, Zhihai Liu, Yan Li, and Renxiao Wang. Comparative assessment of scoring functions: the casf-2016 update. *Journal of chemical information and modeling*, 59(2):895–913, 2018.

[269] Marcel L Verdonk, Jason C Cole, Michael J Hartshorn, Christopher W Murray, and Richard D Taylor. Improved protein–ligand docking using gold. *Proteins: Structure, Function, and Bioinformatics*, 52(4):609–623, 2003.

[270] Santiago Vilar, Giorgio Cozza, and Stefano Moro. Medicinal chemistry and the molecular operating environment (moe): application of qsar and molecular docking to drug discovery. *Current topics in medicinal chemistry*, 8(18):1555–1572, 2008.

[271] Richard A Friesner, Jay L Banks, Robert B Murphy, Thomas A Halgren, Jasna J Klicic, Daniel T Mainz, Matthew P Repasky, Eric H Knoll, Mee Shelley, Jason 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.

[272] Thomas A Halgren, Robert B Murphy, Richard A Friesner, Hege S Beard, Leah L Frye, W Thomas Pollard, and Jay L Banks. Glide: a new approach for rapid, accurate docking and scoring. 2. enrichment factors in database screening. *Journal of medicinal chemistry*, 47(7):1750–1759, 2004.

[273] Qurrat Ul Ain, Antoniya Aleksandrova, Florian D Roessler, and Pedro J Ballester. Machine-learning scoring functions to improve structure-based binding affinity prediction and virtual screening. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 5(6):405–424, 2015.

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

[275] Evan N Feinberg, Debnil Sur, Zhenqin Wu, Brooke E Husic, Huanghao Mai, Yang Li, Saisai Sun, Jianyi Yang, Bharath Ramsundar, and Vijay S Pande. Potentialnet for molecular property prediction. *ACS central science*, 4(11):1520–1530, 2018.

[276] Renxiao Wang, Xueliang Fang, Yipin Lu, Chao-Yie Yang, and Shaomeng Wang. The pdbbind database: methodologies and updates. *Journal of medicinal chemistry*, 48(12):4111–4119, 2005.

[277] 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.

[278] Wei Lu, Qifeng Wu, Jixian Zhang, Jiahua Rao, Chengtao Li, and Shuangjia Zheng. Tankbind: Trigonometry-aware neural networks for drug-protein binding structure prediction. *Advances in neural information processing systems*, 35:7236–7249, 2022.

[279] David Mendez, Anna Gaulton, A Patrícia Bento, Jon Chambers, Marleen De Veij, Eloy Félix, María Paula Magariños, Juan F Mosquera, Prudence Mutowo, Michał Nowotka, et al. Chembl: towards direct deposition of bioassay data. *Nucleic acids research*, 47(D1):D930–D940, 2019.

[280] Davide Chicco and Giuseppe Jurman. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, 21:1–13, 2020.

[281] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[282] Stephane Betzi, Riazul Alam, Mathew Martin, Donna J Lubbers, Huijong Han, Sudhakar R Jakkaraj, Gunda I Georg, and Ernst Schoönbrunn. Discovery of a potential allosteric ligand binding site in cdk2. *ACS chemical biology*, 6(5):492–501, 2011.

[283] Krishna Gade, Sahin Cem Geyik, Krishnaram Kenthapadi, Varun Mithal, and Ankur Taly. Explainable ai in industry. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3203–3204, 2019.

[284] Damien Garreau and Ulrike Luxburg. Explaining the explainer: A first theoretical analysis of lime. In *International conference on artificial intelligence and statistics*, pages 1287–1296. PMLR, 2020.

[285] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[286] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[287] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017.

[288] Xiao-Hui Li, Yuhan Shi, Haoyang Li, Wei Bai, Yuanwei Song, Caleb Chen Cao, and Lei Chen. Quantitative evaluations on saliency methods: An experimental study. *arXiv preprint arXiv:2012.15616*, 2020.

[289] Ameen Ali, Thomas Schnake, Oliver Eberle, Grégoire Montavon, Klaus-Robert Müller, and Lior Wolf. Xai for transformers: Better explanations through conservative propagation. In *International conference on machine learning*, pages 435–451. PMLR, 2022.

[290] Jesse Vig. A multiscale visualization of attention in the transformer model. *arXiv preprint arXiv:1906.05714*, 2019.

[291] Gowri Nayar, Alp Tartici, and Russ B Altman. Paying attention to attention: High attention sites as indicators of protein family and function in language models. *bioRxiv*, pages 2024–12, 2024.

[292] Sarthak Jain and Byron C Wallace. Attention is not explanation. *arXiv preprint arXiv:1902.10186*, 2019.

[293] Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. *arXiv preprint arXiv:1908.04626*, 2019.

[294] Jun Zou, Huan-Zhang Xie, Sheng-Yong Yang, Jin-Juan Chen, Ji-Xia Ren, and Yu-Quan Wei. Towards more accurate pharmacophore modeling: Multicomplex-based comprehensive pharmacophore map and most-frequent-feature pharmacophore model of cdk2. *Journal of Molecular Graphics and Modelling*, 27(4):430–438, 2008.

[295] Tahir Ali Chohan, Jiong-Jiong Chen, Hai-Yan Qian, You-Lu Pan, and Jian-Zhong Chen. Molecular modeling studies to characterize n-phenylpyrimidin-2-amine selectivity for cdk2 and cdk4 through 3d-qsar and molecular dynamics simulations. *Molecular BioSystems*, 12(4):1250–1268, 2016.

[296] Tahir Ali Chohan, Hai-Yan Qian, You-Lu Pan, and Jian-Zhong Chen. Molecular simulation studies on the binding selectivity of 2-anilino-4-(thiazol-5-yl)-pyrimidines in complexes with cdk2 and cdk7. *Molecular BioSystems*, 12(1):145–161, 2016.

[297] Martin Buttenschoen, Garrett M Morris, and Charlotte M Deane. Posebusters: Ai-based docking methods fail to generate physically valid poses or generalise to novel sequences. *Chemical Science*, 15(9):3130–3139, 2024.

[298] Ramanathan Natesh, Sylva LU Schwager, Edward D Sturrock, and K Ravi Acharya. Crystal structure of the human angiotensin-converting enzyme–lisinopril complex. *Nature*, 421(6922):551–554, 2003.

[299] Joseph L Izzo Jr and Matthew R Weir. Angiotensin-converting enzyme inhibitors. *The Journal of Clinical Hypertension*, 13(9):667, 2011.

[300] Maciej Wójcikowski, Pedro J Ballester, and Pawel Siedlecki. Performance of machine-learning scoring functions in structure-based virtual screening. *Scientific Reports*, 7(1):46710, 2017.

[301] Michael M Mysinger, Michael Carchia, John J Irwin, and Brian K Shoichet. Directory of useful decoys, enhanced (dud-e): better ligands and decoys for better benchmarking. *Journal of medicinal chemistry*, 55(14):6582–6594, 2012.