Machine Learning Scoring Functions to Improve Molecular Docking Against Protein-Protein Interaction Targets

Sumin Park

Thesis submitted to the University of Ottawa in partial
fulfillment of the requirements for the degree of
Master of Science in Chemistry

Department of Chemistry and Biomolecular Sciences
Faculty of Science
University of Ottawa

uOttawa

Abstract

Identification of novel therapeutic agents to modulate disease-specific protein targets has been a successful strategy in modern-day drug discovery. While classical targets are receptors, enzymes, and ion channels, protein-protein interaction (PPI) targets are gaining popularity in recent years. PPIs regulate cellular mechanisms associated with vital life processes including signal transduction, cell proliferation, growth, differentiation, and apoptosis. While there are more than 650,000 reported PPIs in the human interactome, only a small fraction of them have been targeted and developed into clinically available drugs. The scarcity of PPIs as biological targets in the drug market derives from significant challenges posed by the structural and topological characteristics of PPI interfaces, which are expansive, flat, and hydrophobic compared to well-defined pockets of conventional binding sites.

To overcome these challenges, computational methods such as structure-based virtual screening (SBVS) have been applied to accelerate the discovery of small-molecule PPI modulators. SBVS utilizes molecular docking simulations to estimate binding affinity and screens large compound libraries to identify virtual hits. In the last decade, scoring functions (SFs), a major component of docking that evaluates the binding energy and pose of a given ligand, have made the transition from being physics-based to machine learning (ML)-based. Numerous studies indicate that machine learning scoring functions (MLSFs) perform better or at least comparably to physics-based SFs, driving the development of a wide variety of MLSFs over the past decade.

In this work, we present new benchmarking datasets and MLSFs tailored PPI targets, designed to improve pose selection in molecular docking. To train and evaluate MLSFs for the drug discovery of PPI targets, we constructed a database consisting of PPI inhibitor poses docked into binding pockets via re-docking and cross-docking with AutoDock and GNINA. Benchmarking this database for the docking power—the ability to identify near-native binding poses—revealed significant room for improvement in pose prediction. The PPI databases were used to train and cross-validate ML models using a variety of interaction-based 3D features, and architectures ranging from shallow models to graph neural networks (GNNs). Our best performing GNN models outperformed two state-of-the-art MLSFs, GNINA and PIGNet2, demonstrating the effectiveness of utilizing interaction features (rather than atomic or molecular-level descriptors) and graph architectures on non-biased datasets. Our work enables fair evaluation of MLSFs using diverse, realistic docking scenarios and introduces a novel computational strategy for identifying small-molecule PPI inhibitors through virtual screening, paving the way for prospective pharmacological investigations of these challenging targets.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor Francesco Gentile, for his guidance and support throughout this journey. Without his unwavering encouragement and trust, none of this would have been possible. Thank you for your genuine kindness, empathy and generosity, I feel very honored to be one of your first graduate students. Your mentorship taught me lessons I'll carry with me for a long time, both professionally and personally.

I would like to thank the current members of our group, who have always been supportive and kind, and never failed to create a lighthearted environment in stressful situations. As coworkers who share a small office and see one another most days, thank you for tolerating me for this long and being a good sport about my constant complaints, broken sense of humor, and other schemes.

I would like to thank my fellow MSc student Stasa Skorupan, for being a fellow guinea pig for the past two years, for never saying no to my schemes (whether sane or not), without each other I don't think we would have ~~survived~~ come this far. Thank you for always reminding me of deadlines and birthdays, for your emotional support and for coming up with the best nicknames. I would also like to thank our postdoc Rahul Ravichandran, for introducing me to the field of computational medicinal chemistry when I first joined the lab. Thank you for always putting up with our shenanigans, for being the alpha donkey of the lab, and most importantly for your priceless quotes, there isn't enough space on the whiteboards in the entire STEM building to write them all. To our former honors student and now MSc student, Kaitlyn Bessette, who must be credited for some of the docking simulations performed in this work, thank you for your support and company, and we hope to have you back in the lab soon. To our first and only PhD student, Thanawat Thaingtamtanha, thank you for your IT skills and German lessons, and for always having time to troubleshoot issues for us. Without your mantis and ant we won't be the Gentile lab. To our research assistant Yasaman Shahrasbi, thank you for your kind words and encouragement, your calm and gentle nature is much appreciated in the lab. I would also like to thank many other past members of the Gentile lab, whether for their academic support, advice, or camaraderie.

To my family, I would like to thank them for their limitless support, love, and trust. While it is not a small sacrifice to be so far away from them to complete my education, I consider myself incredibly fortunate to have a family who are so proud of me, regardless of what I achieve.

Last, I would like to thank everyone in my community, whether friends, acquaintances or teachers, who have made me feel supported and at home no matter how far or close they are. Though challenging, the last two years have been deeply enriching, and I wouldn't be who I am today without them.

Contents

List of Figures

List of Tables

List of Abbreviations

- **ADMET** Adsorption, Distribution, Metabolism, Excretion, and Toxicity
- **AChE** Acetylcholinesterase
- **AI** Artificial Intelligence
- **AUC** Area Under Curve
- **BAP** Binding Affinity Prediction
- **BSA** Buried Surface Area
- **CASF** Comparative Assessment of Scoring Functions
- **CASP13** Critical Assessment of Protein Structure Prediction 13
- **CCV** Clustered Cross-Validation
- **CNN** Convolutional Neural Network
- **Cryo-EM** Cryo-Electron Microscopy
- **CS** Computational Solvent
- **CSAR** Community Structure Activity Resource
- **DBVS** Docking-Based Virtual Screening
- **DDD** Drug Discovery and Development
- **DEKOIS** Demanding Evaluation Kits for Objective In Silico Screening
- **DL** Deep Learning
- **DNN** Deep Neural Network
- **DUD** Directory of Useful Decoys
- **DUD-E** Directory of Useful Decoys-Enhanced
- **FBLD** Fragment-Based Lead Discovery
- **FBDD** Fragment-Based Drug Design
- **GAT** Graph Attention Network
- **GCN** Graph Convolutional Network
- **GBDT** Gradient Boosting Decision Tree
- **GIN** Graph Isomorphism Network
- **GNN** Graph Neural Network
- **GPU** Graphic Processing Unit
- **HPO** Hyperparameter Optimization
- **HTS** High-Throughput Screening
- **IPA** Interaction Pseudo Atom
- **KG** Knowledge Graph
- **LBDD** Ligand-Based Drug Discovery
- **LBVS** Ligand-Based Virtual Screening
- **ML** Machine Learning
- **MLP** Multilayer Perceptron
- **MLSF** Machine Learning Scoring Function
- **MOE** Molecular Operating Environment (software)
- **MD** Molecular Dynamics
- **MIEC** Molecular Interaction Energy Component
- **MUV** Maximum Unbiased Validation (dataset)
- **NMR** Nuclear Magnetic Resonance

- **PDB** Protein Data Bank
- **PMF** Potential of Mean Force
- **PPI** Protein-Protein Interaction
- **QSAR** Quantitative Structure-Activity Relationship
- **R&D** Research and Development
- **ReLU** Rectified Linear Unit
- **RF** Random Forest
- **RMSD** Root Mean Square Deviation
- **ROC** Receiver Operating Curve
- **SASA** Solvent-Accessible Surface Area
- **SBVS** Structure-Based Virtual Screening
- **SF** Scoring Function
- **SR** Success Rate
- **SVM** Support Vector Machine
- **SVR** Support Vector Regression
- **TDC** Therapeutic Data Commons
- **TIFP** Triplet Interaction Fingerprint
- **ULVS** Ultra-Large Virtual Screening
- **VS** Virtual Screening
- **XGBoost** Extreme Gradient Boosting

1 Introduction

1.1 Computational Drug Discovery and the Role of Artificial Intelligence

Drug discovery and development (DDD) is a lengthy and costly process, typically taking 10-15 years and costing $1-2 billion US dollars to bring a single drug to market, with only around 10% of candidates that enter clinical trials ultimately succeeding[1,2]. Despite the clinical stages of DDD taking the longest amount of time in this process, up to 43% of the research and development (R&D) expenditure is accounted for by the preclinical studies, including the costs associated with potential drugs that never enter or complete clinical trials[2]. This high failure rate can be attributed to factors originating from early discovery phases, such as invalid target validation and inadequate ligand property optimization regarding efficacy, selectivity, and toxicity[3]. Thus, advancements at the early stages of drug discovery would greatly accelerate and benefit the subsequent efforts in the entire DDD pipeline, facilitating faster and more economical ways to bring a molecule from bench to clinic.

In the past decade, the development and application of various artificial intelligence (AI)/machine learning (ML) techniques have transformed the research landscape of medical sciences, including the field of drug discovery[4–7]. Broadly speaking, three main factors contributed to the emergence and success of AL/ML methods: (1) structural revolution, wherein 3D structures of biological macromolecules are determined via experimental techniques such as X-ray crystallography, nuclear magnetic resonance (NMR), and cryo-electron microscopy (cryo-EM), resulting in massive databases of clinically relevant protein targets with more than 227,000 structures available in the Protein Data Bank (PDB) as of 2025[8] (2) rapid growth of virtual chemical spaces, made available via ultra-large chemical libraries containing made-on-demand compounds that possess drug-like properties, projected

to expand beyond trillions of diverse and novel molecules[9,10] (3) widespread availability of cloud-computing and graphics processing units (GPUs) driving forward memory intensive and GPU-enabled algorithms such as deep learning (DL)[11].

Importantly, the multiple, often iterative (rather than linear) phases of preclinical drug discovery have been infused with ML, including target discovery, hit identification, hit-to-lead, and lead optimization (**Fig. 1.1**).

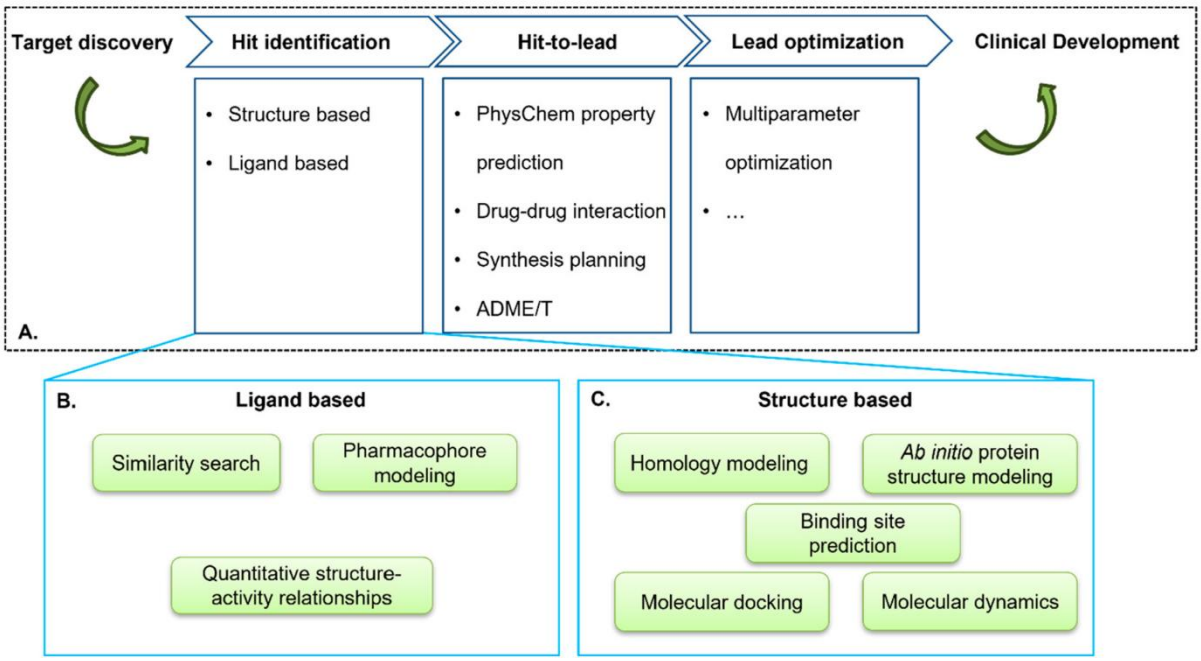


Figure 1.1: Graphic overview of preclinical stages and the role of computational methods at each stage. Target discovery involves identifying and experimentally validating a disease-relevant protein, determining its structure, and elucidating its binding site. Hit identification involves discovering molecules that bind to the target, while hit-to-lead and lead optimization aims to enhance the pharmacokinetic and pharmacodynamic properties of these initial hits. Reprinted from Xia et al. "Integrated Molecular Modeling and Machine Learning for Drug Design," *J. Chem. Theory Comput.* **2023**, 19 (21), 7478–7495. Published by the American Chemical Society. Reprinted with permission.

1.1.1 Target Discovery

A common method for target identification is genomic and phenotypic studies that guide the discovery of a druggable protein implicated in pathogenesis[13]. Often, disease pathways involve a complex network of genes, proteins, mRNAs, and metabolites, tied together by various interactions and associations between individual components[14]. ML

algorithms have shown excellent performance in predicting links or relationships in such complicated networks, and have been implemented in disease target identification in the form of knowledge graphs (KGs)[15,16]. KG is a graph-structured data model containing nodes that represent diseases, genes, proteins, phenotypes, and other heterogeneous types of data, connected by edges representing different relationships or interactions between the nodes[16]. ML models trained on KGs can predict new edges that indicate a relationship between certain diseases and protein targets.

Following target identification, the next step is structure determination, as it is well-established that the function of a protein is governed by its tertiary structure, or folding[17]. Despite the structural revolution discussed earlier, experimentally solving a protein's structure remains costly and time-consuming—and in some cases, it may be impossible to obtain a structure or capture it in its biologically relevant conformation, such as when bound to a ligand. To overcome this challenge, AI-based sequence-to-structure methods were developed, which are currently one of the most active areas of research in applications of AI/ML in structural biology. In 2018's critical assessment of protein structure prediction 13 (CASP13), AlphaFold was introduced by DeepMind, a convolutional neural network (CNN) model trained on PDB to predict pairwise residue distances and torsion angles to accurately determine the fold topology of a protein, and significantly outperformed other state-of-the-art computational methods[18]. DeepMind published two subsequent versions, AlphaFold 2 and AlphaFold 3, which demonstrated improved accuracy in structure prediction in CASP14[19] and applicability in specialized areas such as protein-ligand interactions, protein-nucleic acid interactions, and antibody-antigen predictions[20], respectively. Fueled by the breakthrough in structural biology achieved with AlphaFold models, several other DL-based methods for structure prediction were developed in recent years. AlphaFold-Multimer and AF2Complex

build upon AlphaFold2 to improve accuracy in predicting multi-chain protein complexes[21,22]. ESMFold is a fast sequence-to-structure predictor that achieves approximately 60x faster performance than AlphaFold2 while maintaining comparable accuracy, leveraging a transformer-based protein language model[23]. Most recently, Boltz-1 was developed by Wohlwend et al., the first fully open-source structure prediction model that performs on par with AlphaFold 3 and other state-of-the-art models[24].

Last but not least, binding site elucidation is a crucial task in the early stages of drug discovery, as protein function depends on its interaction with other biological molecules, commonly known as a ligand[25]. While conventional physics-based methods are available, recent years have seen the emergence of several ML-based models designed to identify druggable binding sites suitable for small-molecule ligands on protein surfaces. DeepSite was trained on the sc-PDB[26] database and employs 3D descriptors and a CNN architecture for binding pocket prediction[27]. PUResNet utilizes a deep residual neural network and exhibits superior performance compared to DeepSite[25]. Beyond conventional protein-ligand binding sites, identification of hard-to-detect pockets known as cryptic pockets expands the horizon of druggable targets[28]. PocketMiner is a graph neural network (GNN) trained for efficient prediction of the cryptic pockets that open during MD simulations. The model is trained to identify residues contributing to the cryptic pocket formation from a static 3D structure, and has been applied to scan the entire human proteome to identify new cryptic pockets[28].

1.1.2 Hit Identification

The identification of the potential compounds that bind and modulate the function of a target protein is the most important step of early-stage drug discovery, as subsequent hit-to-lead and lead optimization efforts depend on the quality of the discovered hits.

1.1.2.1 Ligand-based Drug Discovery and Virtual Screening

Ligand-based drug discovery (LBDD) leverages the knowledge of molecules that bind to a target of interest to improve existing drugs or develop new drugs with better pharmacological properties[29]. The advantage of LBDD is that it doesn't require knowledge of the target's structure; instead, it relies on the features—structural or otherwise—of known ligands to build predictive models of bioactivity for new compounds. Quantitative structure-activity relationship (QSAR) is a popular paradigm in LBDD that predicts the changes in potency as a function of structural modifications in ligands[30]. Classical QSAR models establish a linear relationship between molecular descriptors and a desired property to predict, ranging from bioactivity to pharmacokinetic properties such as adsorption, distribution, metabolism, excretion, and toxicity (ADMET)[31]. ML-based methods apply a non-linear regression to the QSAR problem to achieve better accuracy using a variety of different algorithms. QSAR models built from ensemble and DL-based algorithms have demonstrated excellent predictive performance across large and diverse datasets[32,33].

Virtual screening (VS) is an *in-silico* strategy to sift through a vast chemical space to identify potential hit molecules by computationally estimating their affinity to a target of interest. VS played a central role in many successful early-stage drug discovery campaigns, proving to be one of the most effective and fast methods at a relatively low cost compared to conventional experimental high-throughput screening (HTS), which suffers from high cost and low hit rate[34]. There are two main branches of VS: ligand-based virtual screening (LBVS), which relies on the structure and properties of known ligands to identify new binders, and structure-based virtual screening (SBVS), which takes into account both the protein and ligand structures and their binding interaction[35]. Despite recent advancements in ML, LBVS still tends to result in hit molecules that are similar to existing ligands structurally

and property-wise due to the nature of the method, which aims to mimic the template ligand's binding mechanism[36,37].

However, the goal of a drug discovery campaign is often to identify molecules with novel scaffolds that differ significantly from known drugs or to identify a collection of molecules with diverse scaffolds[38]. Hence, SBVS methods can predict the binding conformation and the strength of binding between ligand and target receptor via scoring functions. Docking-based virtual screening (DBVS) is most widely used as it enables the screening of large chemical libraries in a reasonable timeframe without compromising the hit rate[35]. Docking programs aim at predicting two main components of the binding mechanism: the binding mode of a ligand, which is determined by the location, orientation, and conformation of the ligand in complex with the protein, and the free energy of the binding for each mode, generally related to the binding affinity of the ligand[39]. In the VS setting, docking requires a high-resolution 3D structure of the receptor as well as a commercially available compound library (**Fig. 1.2**).

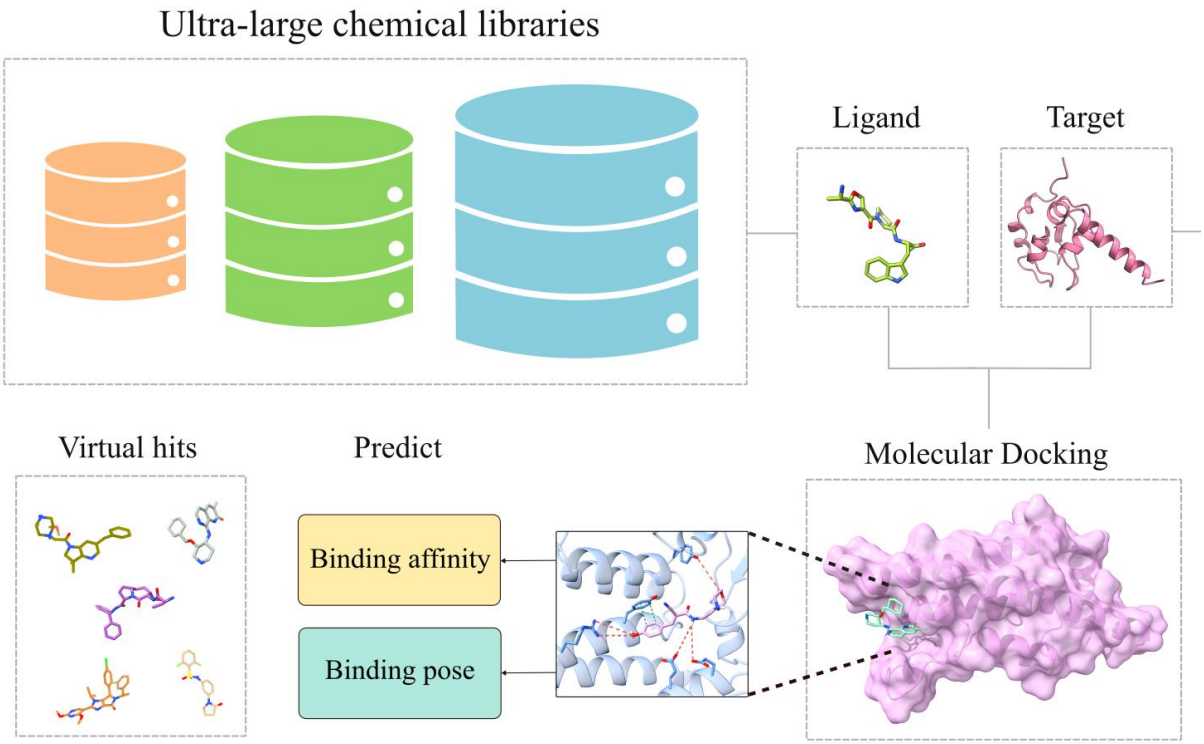


Figure 1.2: Graphical representation of SBVS and the role of molecular docking.

While docking is fast and sufficiently accurate for large-scale virtual screenings as a starting point, it has significant limitations. One of the main drawbacks is that protein structures used in docking are static screenshots of a protein, which fail to capture the dynamic nature of macromolecules in solution. In reality, proteins undergo conformational changes during ligand binding, which docking does not account for. This hinders accurate simulation of the binding when (1) the binding site conformation is not compatible with the docked ligand (2) there are allosteric sites or cryptic pockets on the protein that become apparent only upon ligand interaction[38]. Additionally, docking algorithms struggle with modeling metal ions or solvent effects at the binding site, leading to difficulties with metalloproteins and protein-protein interaction sites which are often flat and solvent exposed[34,38,40]. To overcome such limitations, specialized docking approaches exist, such as ensemble docking, use of constraints, and flexible receptor docking. Post-processing methods, including rescoring with physics-based or ML-based methods, can further refine

results[35]. However, many of these issues can be implicitly addressed by using more sophisticated computational techniques based on molecular dynamics (MD) simulations. MD can provide thermodynamic—and sometimes even kinetic— insights of ligand binding events, revealing binding pathways, energy barriers, and metastable states that docking entirely overlooks. Despite its advantages, MD is computationally expensive, often thousands of times slower than docking, and still requires a starting reliable binding pose. As a result, it is typically used for cryptic pocket detection, protein structure refinement before docking, or small scale post-processing steps such as rescoring and in silico binding assays after DBVS[35].

1.1.2.2 Machine Learning-Augmented Docking

While docking is a fast method compared to MD, docking campaigns are generally not scalable to libraries exceeding hundreds of millions of compounds, especially in the absence of elite computational resources[39]. It is also not clear if docking every single compound in a library can be avoided, as only a tiny fraction of them are true actives, while the rest of the library is discarded. As the scope of the known chemical universe is projected to expand rapidly in the near future, ML approaches to ultra-large virtual screening (ULVS) have been proposed to alleviate the heavy costs of brute-force docking[41]. One way to incorporate ML to accelerate ULVL is by training a ligand-based docking score estimator on a fraction of the library that can be docked within reasonable resources, and inferring the docking scores of the rest of the library to retrieve final hits[39,42]. Lean-Dock employed this idea by training a support vector regression (SVR) model to predict docking scores from molecular fingerprints, achieving a four-fold reduction in docking while retaining a similar number of true actives in the top-scoring docked molecules[43]. Deep Docking expanded this concept via an iterative active learning workflow, in which a small fraction of the library is docked to train a deep neural network (DNN)-based QSAR model at each iteration to reduce

the size of the library. Deep Docking showed a 100-fold reduction in costs while retaining 90% of the top virtual hits in ZINC15 library containing 1.36 billion molecules against a set of 12 targets[44]. This method was prospectively applied to screen approximately 40 billion molecules against SARS-CoV-2 main protease (Mpro) target, from which the top hits were experimentally validated to be active[45].

Another application of ML in docking methods lies in replacing the scoring function and/or the pose generation algorithm with data-driven ML models. In the first approach, docking is performed using a hybrid method, where traditional sampling algorithms generate docked poses, but these are scored using machine learning scoring functions (MLSFs) instead of physics-based ones[10]. The potential and limitations of this synergetic combination will be explored in detail in Section 1.3. More recently, generative ML techniques have been introduced to directly generate ligand poses, aiming to fully replace conventional docking algorithms[39]. While some studies have reported improved performance over traditional methods on specific benchmarks, concerns have been raised about the physical plausibility of the poses produced by these generative approaches[46].

1.2 Protein-Protein Interactions

1.2.1 Protein-Protein Interactions as an Emerging Class of Drug Targets

Protein-protein interactions (PPIs) are responsible for a wide range of biological functions, in which the formation of a complex between two proteins triggers a cascade of cellular mechanisms associated with the regulation of pivotal life processes such as signal transduction, cell proliferation, growth, differentiation, apoptosis, and much more[47,48]. Hence the PPI interactome, or the complex network of proteins, is deeply implicated in many diseases including cancer, infectious diseases, and neurological disorders[49]. Although traditional drug targets primarily include canonical targets such as receptors, enzymes, and

ion channels, PPIs have recently emerged as an attractive class of molecular targets due to their vast therapeutic potential, despite significant challenges associated with this class of therapeutics[50]. Identification of novel therapeutic agents to modulate disease-specific protein targets has been a successful strategy in modern-day drug discovery, enabled by the rapidly expanding database of high-resolution crystal structures and larger, more diverse screening libraries spanning a wide chemical space. With growing collections of experimental data and emerging technologies in medicinal chemistry, modulating PPIs offers significant advantages in advancing novel therapeutic solutions, including but not limited to slowing disease progression, treatment of refractory diseases, and curative therapies[49]. While severely underexplored, it is estimated that there are more than 650,000 reported disease-relevant PPIs in the human interactome compared to 20,000 protein-coding genes[51]. In addition to the prevalence of PPIs in human disease pathways, their interfaces are often less conserved than conventional protein-ligand binding sites, granting PPI modulators a greater likelihood of selectivity, which is arguably one of the most important consideration in drug development[52].

Compared to the number of drugs targeting protein-ligand interfaces, far fewer have been developed for PPIs. Indeed, PPI interfaces are uniquely challenging to target due to their characteristics, which are expansive, flat, and hydrophobic, to the point that they were regarded as nearly "undruggable" with a small molecule[47]. First, PPI interfaces are extensive, with the surface area ranging between 1500 and 3000 $Å^2$, much larger compared to that of a typical protein-ligand interface (300 to 1000 $Å^2$)[53]. Second, the contact surfaces of PPIs are flat and lack any grooves or cavities to bind small-molecule ligands[54]. Due to this feature, PPIs generally lack an endogenous ligand at the binding pocket, which is often the starting reference for the search, discovery, and optimization of suitable molecules into lead compounds[55]. Third, the interface is typically hydrophobic, meaning that any potent

modulator has to make hydrophobic contacts over a large surface area, which has consequences for the pharmacokinetics of a drug[56]. All the above topological characteristics of PPI interfaces render it difficult for small molecule-based modulation. In addition, due to the many features of PPIs that differentiate them from other drug targets, established methodologies are less suited for PPI-specific drug discovery. For example, conventional compound libraries used in high throughput screening (either wet-lab based or virtual) usually cover a chemical space optimized for the discovery of binders of conventional pockets, for example by strictly following rules such as Lipinski's Rule of Five (RO5), resulting in low hit rates[57,58].

1.2.2 Current Advances in PPI Drug Discovery

Classical medicinal chemistry approaches have largely failed to overcome the challenges posed by PPIs, highlighting the need for more effective strategies to design and identify PPI modulators. In recent years, the discovery of "hotspot" residues has become a key element of PPI-based drug development. Hotspots are defined as continuous or discontinuous patches of residues that contribute significantly towards the binding free energy[59]. Despite the large surface area of PPI interfaces, only a small number of key amino acid residues play a crucial role in binding, making them ideal targets for ligand design[47]. Hotspot regions are identified through alanine-scanning mutagenesis, established on residues where alanine mutations cause an energy difference of more than 2 kcal/mol, and validated through X-ray crystallography or NMR[60]. The identification of hotspot regions has been a successful strategy to guide the structure-based rational design of PPI modulators. Another common approach is fragment-based drug design (FBDD), sometimes referred to as fragment-based lead discovery (FBLD). In FBDD, chemical fragments (> 200 Da) with weak binding affinity are identified and then linked to other fragments binding to nearby regions, to

design a lead compound with a stronger affinity to the target protein[61,62]. Given the expansiveness of the PPI binding site and discontinuity of hotspot regions, once the fragment hits are identified, these methods have been proposed for the design of PPI-modulating compounds within a reasonable molecular weight range (300-500 Da)[48]. FBDD proved itself to be an effective strategy in few cases, including the discovery of inhibitors of XIAP/caspase-9, Bcl-2/Bax, and bromodomains PPIs[48]. Another structure-based approach is peptidomimetic design, relying on the design of modulators mimicking the secondary structure of the interacting domain of one of the two PPI partners[53]. Peptidomimetics can be non-peptidic molecules that match the structural motives of the peptides, or small peptides with various modifications to improve their metabolic stability[53]. Often, peptides by themselves struggle to fold into their bioactive conformation or stay in the folded shape for a desirable timespan, thus requiring alterations such as incorporating unnatural amino acids, N-methylation (backbone modification), and the use of foldamers[63]. Computational methods can be incorporated at different stages of the aforementioned strategies, both to accelerate the process and to explore new ways of tackling challenging PPI targets. For example, the computational solvent (CS) mapping technique is used to identify "druggable" sites on the protein surface by simulating fragment-sized chemical probes around the protein surface and estimating the lowest free energy site[64]. LBVS, which was introduced in Section 1.1.2.1, has also been successfully employed to aid the development PPI modulators such as Ubc13/Uevl[65], MDM2/p53[66], and TCF/β-catenin[67].

While some advances made in PPI drug development shifted the focus from small molecule-based ligands to larger scaffolds such as peptides, antibodies, and macrocycles, which better align with requirements (i.e. shape complementarity and ability to make more hydrophobic contacts) suited to the unique topology of PPI interfaces, the advantages of

small molecule modulators remain significant. Small molecules can be optimized for oral availability, are generally cheaper to synthesize, and can possess more favorable pharmacokinetic profiles than other therapeutic classes[51]. Indeed, the development of larger molecules into clinical candidates is often hindered by poor membrane permeability, inability to target intracellular targets, cost, and synthetic challenges[48,56]. Small molecules have an edge in terms of available drug discovery technologies, as many of the tools were developed with those in mind, as opposed to unusual scaffolds[51].

1.3 Scoring Functions

As briefly discussed in Section 1.1.2.1, molecular docking consists of two parts: pose generation and scoring. In the first step, probable ligand conformations relative to a target protein, i.e. 3D coordinates, orientation, and internal rotations, are generated. In the second step, each pose is assigned a score for its favorable interaction with the target and ranked[38]. Pose generation and scoring are often intertwined, as many modern docking programs use sampling algorithms—such as genetic algorithms or Monte Carlo sampling—wherein the generation of new poses is guided by the docking scores of previous batches to accelerate the process[68]. In the second step, the absolute binding free energy of a given pose is determined by a scoring function (SF), that approximates enthalpic and entropic contributions involved in the binding. A good SF must account not only for the energy contributions from the direct interaction between protein and ligand atoms, but also for solvation and desolvation effects, ligand flexibility, and the overall change in energy from the unbound to the bound state of the complex[40,68]. Since the introduction of the first docking program by Kuntz et al.[69] in 1982, hundreds of SFs have been developed to improve the accuracy of binding energy estimation by incorporating the aforementioned factors, either explicitly or implicitly. SFs can be classified into two broad categories based on their functional form: classical SFs, which

generally use a fixed linear functional form to sum up the interaction energies, and MLSFs, which utilize nonlinear regression to implicitly capture those interactions[70,71]. Physics-based SFs, considered the first generation within the classical SF family, estimate binding energy by summing key non-covalent interactions—such as hydrogen bonding, van der Waals (dispersion) forces, and electrostatic or ionic interactions—between protein-ligand atomic pairs using a force field[72]. However, these methods often neglect entropic contributions and solvent effects, leading to suboptimal accuracy in binding energy estimation[73]. As such, later SFs incorporate terms for entropic penalty for torsion and solvation (via independent solvent models) to enhance performance[34,73]. Empirical SFs are another type of classical SF, and calculate the weighted sum of energetic components, such as van der Waals, hydrogen bonding, hydrophobic effects, and so on[71]. Training empirical SFs is a simple linear regression problem in which the weights of each term are parameterized with protein-ligand complexes and their known binding affinities[71]. Knowledge-based SFs represent the final class within the classical SF category. They rely on statistical thermodynamics to calculate pairwise protein-ligand atom potentials—known as potentials of mean force (PMFs)—trained from the 3D structures of known protein–ligand complexes[74]. These functions offer several advantages: (1) they do not require binding affinity data, relying solely on bound structures, as interatomic distances are sufficient for parameterizing the SF and (2) entropic effects and solvation are implicitly accounted for, as PMFs approximate the Helmholtz free energy, and a volume correction term addresses solvation effects[74]. **Fig. 1.3** summarizes the different types of SFs discussed in this section.

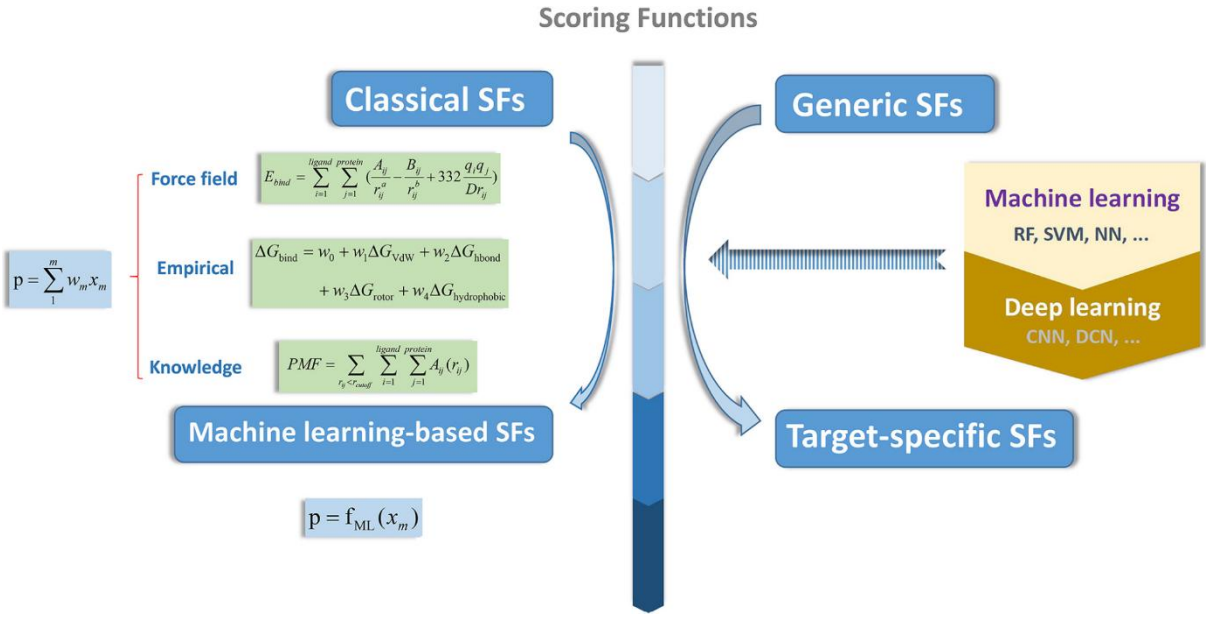


Figure 1.3: Overview of different types of SFs. Reprinted from Chen et al. "From Machine Learning to Deep Learning: Advances in Scoring Functions for Protein–Ligand Docking," *WIREs Comput. Mol. Sci.* **2020**, *10* (1), e1429. Copyright 2020 John Wiley & Sons. Reprinted with permission.

1.3.1 Machine Learning Scoring Functions

1.3.1.1 Comparisons with Conventional SFs

In this section, the performance of classical SFs in screening and scoring-type tasks (as scoring is closely related to docking and ranking) is examined in comparison to MLSFs. MLSFs published as early as 2010 claimed to surpass classical SFs. **Fig. 1.4** illustrates the reported performance of several MLSFs benchmarked on CASF-2007 against a diverse set of classical SFs published between 1998 and 2010, including GOLD::GoldScore[75] (force-field), DrugScore^{CSD}[76] (knowledge-based), X-Score::HMScore[77] (empirical), SYBYL::ChemScore[78] (empirical), GlideScore-XP[79] (empirical), SYBYL::PMF-Score[78] (knowledge-based), and Autodock Vina[80] (semi-empirical).

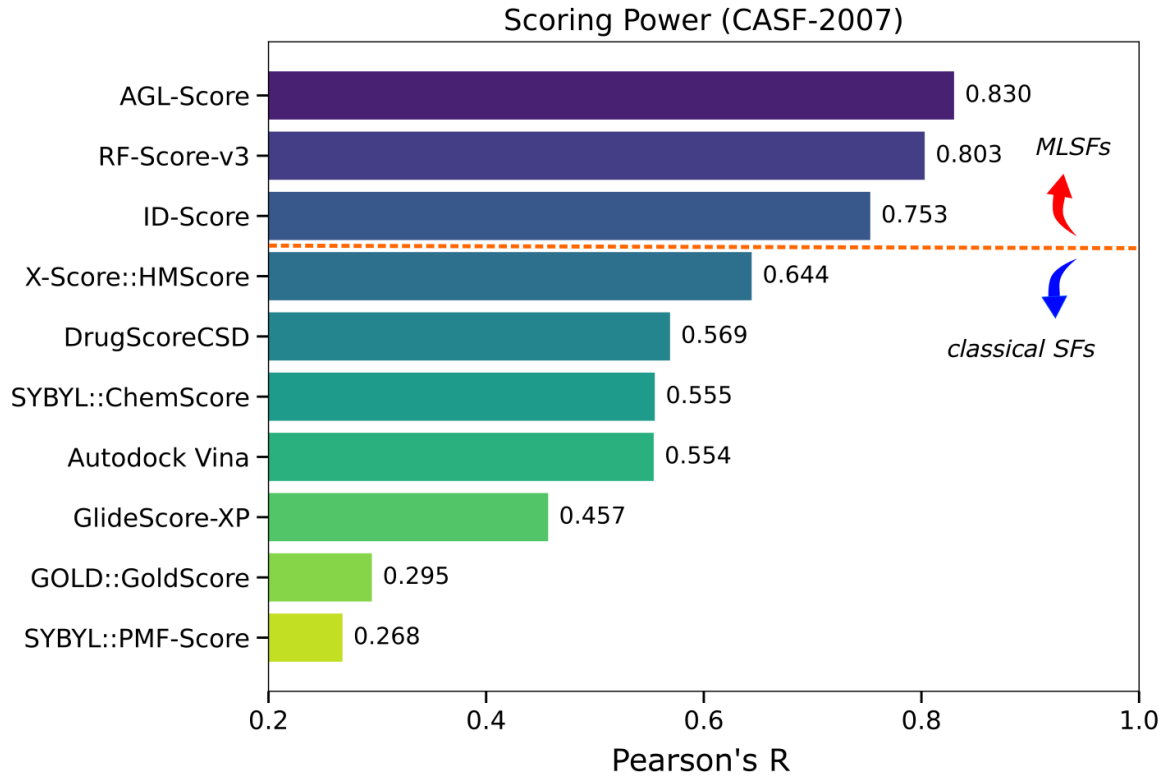


Figure 1.4: Scoring power comparison between MLSFs and classical SFs in terms of Pearson's R value. Pearson's R measures the correlation between actual binding energy and predicted binding scores. Adapted from Nguyen et al. "AGL-Score: Algebraic Graph Learning Score for Protein–Ligand Binding Scoring, Ranking, Docking, and Screening," *J. Chem. Inf. Model.* **2019**, *59* (7), 3291–3304. Copyright 2019 American Chemical Society. Adapted with permission.

On CASF-2007, the most predictive SF in this list, X-Score (R = 0.644)[82], performs worse than three MLSFs shown: RF-Score (R = 0.776)[83], RF-Score-v3 (R = 0.803)[84], AGL-Score (R = 0.830)[81]. The performance gap between ML-based and other types of SFs for binding affinity prediction (BAP) was also shown to be true when benchmarked on other datasets[85,86]. MLSFs trained for SBVS showed strong improvements over classical SFs in terms of EF and AUC across different benchmarks. For example, RF-Score-VS, trained and tested on DUD-E, shows more than a threefold increase in $EF_{1\%}$ when comparing the best performing model to Autodock Vina across diverse targets[87]. A CNN model built by Ragoza et. al. was benchmarked with DUD-E and compared to Autodock Vina, resulting in a 2- to 4-fold increase in $EF_{1\%}$, as well as increase in AUC (Vina: 0.716; CNN: 0.868)[88]. Similar

results were obtained from SIEVE-Score[89] (random forest) and BT-Screen[90] (gradient-boosted decision tree) in which the MLSFs were compared to Glide[79] and X-Score[77] on DEKOIS 2.0. and PDBbind, respectively.

By now, there is sufficient evidence provided by numerous comparative retrospective and prospective studies accumulated over the last two decades to suggest that MLSFs perform better or at least comparably to classical SFs. Although current evaluation tools have known limitations (as will be discussed in the next section), both classical and MLSFs are similarly affected by flaws in benchmarking[70,91], and observed performance gaps often reflect the inherent differences in capacity between traditional methods and ML models. MLSFs were developed to overcome the limitations of classical SFs, which utilize a predefined linear functional form that sums up enthalpic (and in certain cases, entropic) contributions calculated from the interaction between the ligand atoms and the protein residues at the binding site[70–72]. The first issue lies in the additive nature of physicochemical terms in the linear formulation of classical SFs, which fails to capture the complex interplay between enthalpic and entropic contributions. This can lead to redundancy between terms or leave portions of the binding energy unaccounted for[40,92]. The second issue involves several complex components—such as solvation and entropy—which are often highly approximated. Despite ongoing efforts to better model these contributions, they remain a major bottleneck in the development of more accurate SFs due to the complexity of the underlying physics and the inherently static nature of docking[34]. On the other hand, MLSFs utilize data-driven algorithms to map patterns in experimental protein-ligand complexes to scoring, docking, ranking, and screening predictions. The non-linearity of this method frees the SFs from the constraints of the linear functional form by implicitly accounting for protein-ligand binding interactions without depending on the assumptions of a predefined form[71,72].

1.3.1.2 Overview of MLSFs

In the past decade and a half, dozens of MLSFs have been developed and implemented in SBVS campaigns to successfully discover new hit molecules. MLSFs can be broadly classified into traditional ML techniques and those that come under the umbrella of emerging DL techniques. In the former category, random forest (RF) is a popular ensemble algorithm that combines multiple decision tree predictors[93]. RF is employed in RF-Score[83] (and its subsequent versions, RF-Score-v2[94], RF-Score-v3[84]), RF-Score-VS[87], Δ_{vina}RF20[95], and SIEVE-Score[89]. Prospectively, RF-Score (while designed for scoring) was used in anti-bacterial hit identification to successfully discover new inhibitors for type II dehydroquinase targets of *Mycobacterium tuberculosis* and *Streptomyces coelicolor*, resulting in 25 (out of 148) hits at $IC_{50} \leq 250$ μM, and 32 (out of 148) hits at $IC_{50} \leq 250$ μM, respectively[96]. Another popular algorithm often utilized for molecular data prediction is support vector machine (SVM), which is a supervised learning method capable of dealing with high-dimensional data for classification[97]. SVM was used in many target-specific MLSFs for screening, such as PESD-SVM[98], SVR-KBD[86], MIEC-SVM[99], SVMGen[100], and PLEIC-SVM[101]. Out of these, MIEC-SVM is a kinase-specific SF with molecular interaction energy components (MIECs) as its features, which has also been prospectively used in SBVS to discover 7 actives (out of 50) for ALK kinase target at $IC_{50} < 10$ μM[82]. While SVM was originally developed for classifiers, its derivative SVR can be used for scoring tasks, such as SVR-KB and SVR-EP based on knowledge-based pairwise potentials and physicochemical properties as features, respectively[86]. Last but not least, gradient boosting decision tree (GBDT)[102], or its scalable version, extreme gradient boosting (XGBoost)[103], are two other ensemble algorithms that are widely implemented in MLSFs, with examples such as FFT-BP[104], BT-Score/BT-Screen/BT-Dock[90], ΔvinaXGB[105], XGB-Score[106], AGL-Score[81] (discussed earlier in comparison to classical SFs), and vScreenML[107]. vScreenML was prospectively evaluated to screen for the

human acetylcholinesterase (AChE) target, resulting in 10 (out of 23) compounds at $K_i < 50$ μM[107].

With the emergence of DL algorithms in drug discovery, several DL-based SFs have been recently proposed. CNNs have been widely employed in SFs for BAP and VS by taking the structural information of the protein-ligand complex at the binding site as input data in a voxelized form[72]. In general, the input layer in CNN receives the spatial information at the binding site in the form of 3D grids of predefined volume, with each of the grid cells containing features as simple as atom-type enumeration to more sophisticated protein-ligand interaction descriptors[108–111]. The first CNN-based SF, AtomNet, was developed in 2015 and successfully validated in a prospective study, identifying 3 (out of 11) compounds at 250 μM for Miro1, a pharmacodynamic marker for Parkinson's disease[112]. Since then, a steady stream of newly developed SFs has emerged, incorporating diverse CNN architectures and feature representations to improve performance. DeepVS[113] and DenseFS[114] are 3D CNN models utilizing basic structural features such as atom types, charges, distances, and amino acid types for virtual screening. For scoring, a wide variety of features were selected to develop SFs such as DeepDTAF[109], K_{DEEP}[115], Pafuncy[116], OnionNet[110], DeepAtom[108], DeepDTA[117]. GNINA[118,119] is an ensemble CNN SF designed to rescore poses generated from SMINA docking program (which is a version of the popular docking tool Autodock Vina). While it was mainly developed for pose classification, GNINA showed excellent performance applied in SBVS as well[120]. It is important to note that while CNNs have been successfully employed in superior-performing SFs, their training on 3D complexes is computationally costly since it requires different orientations of the structure as input to account for translation and rotational permutations[121]. More recently, GNNs have emerged as state-of-the-art models in DL-based SFs. In GNN models, the protein ligand binding site is represented as a molecular

graph, in which nodes represent atoms and edges are constructed between covalent and/or non-covalent interactions[122]. GNNs are typically permutation invariant, and it is relatively easy to guarantee invariance or equivariance to roto-translations[121]; hence, they are well suited to deal with unordered 3D biomolecular structures of variable size such as protein-ligand complexes. GraphDelta[123] is a multitask GNN that predicts binding affinities (K_d, K_i, and IC_{50}) based on node and edge features that approximate electron energy. PotentialNet[124] is a graph convolutional network (GCN) trained to predict binding affinity based on basic structural features such as atom types, atomic distances, and bonds. Graph convolutional architectures were also implemented in GraphBAR[125] and SIGN[126], both of which are binding affinity estimators that rely on their unique network design to predict the interaction between protein and ligand molecule. PIGNet is a gated graph attention network (GAT) trained for scoring and screening, which predicts the atom-atom pairwise interactions with physics-informed equations parametrized by neural networks[127,128]. Two recent SFs can easily be integrated into existing docking programs for rescoring purposes: RMTScore[129], which predicts correct ligand binding poses by learning the distance likelihood between binding pocket residues and ligand heavy atoms, and EquiScore[130], which estimates docking scores from equivariant heterogeneous graph architecture, incorporating various physical and prior knowledge about intermolecular interaction. Finally, FlexPose[131] and KarmaDock[132] are GNN-based docking tools that fully replace conventional docking methods, including pose generation. These approaches have demonstrated superior performance compared to other widely used MLSFs in docking and screening tasks[46]. **Table 1.1** summarizes the MLSFs discussed in this section, highlighting key details such as the training datasets, benchmark datasets, and data splitting methods between training and test sets.

Table 1.1: Major MLSFs developed to date.

Scoring Function	ML Model	Task	Training Dataset	Benchmark Dataset	Data Split
RF-Score	RF	Scoring	PDBbind v.2007 refined	PDBbind v.2007 core	Overlap removed
RF-Score-v2	RF	Scoring	PDBbind v.2007 refined	PDBbind v.2007 core	Overlap removed
RF-Score-v3	RF	Scoring	PDBbind v.2012 refined	PDBbind v.2007 core, PDBbind v.2013 (released after v.2012)	PDBbind timesplit
RF-Score-VS	RF	Screening	DUD-E	DUD-E, DEKOIS2.0	CV (target-based clustering)
$\Delta_{vina}RF_{20}$	RF	Scoring, Ranking, Docking, Screening	PDBbind v.2004, PDBbind v.2013, CSAR (decoy)	CASF-2007, CASF-2013	Overlap removed
SIEVE-Score	RF	Screening	DUD-E	DUD-E, DEKOIS2.0	CV (per-target evaluation)
PESD-SVM	SVM	Target-specific Screening	PDBbind v.2005 refined	PDBbind v.2005 core	Overlap removed
SVR-KB/SVR-EP	SVM	Scoring	CSAR, PDBbind v.2010 refined+core	CSAR	CSAR SET1 / SET2
SVR-KBD	SVM	Target-specific Screening	DUD	DUD	Unknown
MIEC-SVM	SVM	Target-specific Screening	BindingDB, ZINC	BindingDB, ZINC	Clustered with Tanimoto similarity index
SVMGen	SVM	Screening	sc-PDB v.2012	DUD-E, SARfari	Unknown
PLEIC-SVM	SVM	Target-specific Screening	DUD-E	DUD-E	Random CV
FFT-BP	GBDT	Scoring	PDBbind v.2015 refined	PDBbind v.2007 core	Overlap removed
BT-Score/BT-Dock/BT-Screen	XGBoost	Scoring, Docking, Screening	PDBbind v.2014 refined	PDBbind v.2014 core	CV (target- and ligand-based clustering)
AGL-Score	GBDT	Scoring	CASF-2007, CASF-2013,	CASF-2007, CASF-2013,	Overlap removed

			CASF-2016	CASF-2016	
XGB-Score	XGBoost	Scoring	PDBbind v.2007 refined	PDBbind v.2007 core	Overlap removed
Δ_{vina}XGB	XGBoost	Scoring, Ranking, Docking, Screening	PDBbind v.2016 refined+general, CSAR	CASF-2013 + CASF-2016 (released after PDBbind v.2015)	PDBbind timesplit
vScreenML	XGBoost	Screening	D-COID	DEKOIS2.0, PPI	Overlap removed
AtomNet	CNN	Screening	ChEMBL-20 PMD (in-house DUD-E like dataset from ChEMBL)	ChEMBL-20 PMD	Cluster with Bemis-Murcko scaffold
DeepVS	CNN	Screening	DUD (partial charge corrected version)[133]	DUD	Leave-one-out CV (target-based clustering)
DenseFS	CNN	Screening	DUD-E	DUD-E, independent test set from ChEMBL	CV (target-based clustering)
DeepDTA	CNN	Scoring	Davis, KIBA	Davis, KIBA	Random CV
K_{DEEP}	CNN	Scoring	PDBbind v.2016 refined	PDBbind v.2016 core, CSAR	Overlap removed
Pafuncy	CNN	Scoring	PDBbind v.2016 refined+general	PDBbind v2016 core, Astex Diverse Set	Overlap removed
OnionNet	CNN	Scoring	PDBbind v.2016 general+refined	PDBbind v.2016 core	Overlap removed
DeepAtom	CNN	Scoring	PDBbind v.2016 refined+general, BindingMOAD, Astex Diverse Set	PDBbind v.2016 core, Astex Diverse Set	Overlap removed
DeepDTAF	CNN	Scoring	PDBbind v.2016 general+refined	PDBbind v.2016 core	Overlap removed
GNINA	CNN	Scoring, Docking,	PDB (Pocketome	PDBbind v.2019,	CV (target- and ligand-based)

		(Screening)	v17.12[134])	cross-docking dataset by Wierbowski et al.[135]	
PotentialNet	GNN	Scoring	PDBbind v.2007 refined	PDBbind v.2007 refined, core	CV (target- and ligand-based clustering)
graphDelta	GNN	Scoring	PDBbind v.2018	PDBbind v.2016 core, CSAR NRC-HiQ, CSAR12, CSAR14	Overlap removed
SIGN	GNN	Scoring	PDBbind v.2016 general, refined	PDBbind v.2016 ore, CSAR NRC-HiQ	Overlap removed
graphBAR	GNN	Scoring	PDBbind v.2016 general, refined	PDBbind v.2016 core, PDBbind v.2013 core, CSAR NRC-HiQ	Overlap removed
PIGNet	GNN	Scoring, Screening	PDBbind v.2019 refined	PDBbind v.2016 core, CSAR NRC-HiQ	Overlap removed
RMTScore	GNN	Docking, Screening	PDBbind v.2020 general+refined	PDBbind v.2016 core, DEKOIS2.0, DUD-E	Overlap removed
FlexPose	GNN	Scoring, Docking	PDBbind v.2020, ApoBind	PDBbind v.2020, ApoBind	CV (target-based clustering)
KarmaDock	GNN	Scoring, Docking, Screening	Unknown	ApoBind core, DEKOIS2.0, PDBbind v.2020 core	Unknown
EquiScore	GNN	Screening	PDB, DeepCoy	DUD-E, DEKOIS2.0	Overlap removed

*CV = Cross Validation

1.3.2 Benchmarking SFs in Docking

Since the inception of SBVS, numerous methods have been developed and applied in

early-stage drug discovery. The successful identification of virtual hits, lead compounds, and drugs progressing to clinical trials or market approval offers prospective validation of these methods' effectiveness[136]. However, prospective studies are not only time-consuming and costly, but also impractical for systematically comparing a large number of SFs, as success with one target does not necessarily generalize to others[136,137]. For this purpose, retrospective benchmarking is used, where a series of SFs are evaluated on standardized benchmarking datasets for a specific task. Retrospective benchmarking can guide users' choice in implementing SFs in their projects, as well as identify strengths and weaknesses of an SF to improve current methods[136].

So far, the discussion of the role of SFs in SBVS has been limited to its ability to score protein-ligand complexes, that is, to predict either the absolute binding affinity or binding score that directly correlates to affinity. However, SF can be adapted for other types of tasks, namely docking, ranking, and VS. While all of them are at its core based on docking score, it is an important distinction as each SF can be parametrized and trained in a tailored manner for the specific type of task, regardless of whether it is classical or ML-based. The remainder of this chapter will outline the methods and existing datasets for retrospective benchmarking of SFs for docking, as our project is concerned with improving docking task.

1.3.2.1 Evaluation Metrics

Docking power measures the ability of an SF to determine the correct binding pose that closely resembles the native binding pose of a ligand. The most important metric for docking power is oftentimes success rate (SR), which is calculated by the percentage of docked systems which successfully generated a pose close to a native pose by a certain margin. This margin is determined by root mean square deviation (RMSD) between docked and crystal poses (averaged for the spatial coordinates of each atom in a ligand), as shown in

Eq. 1.1.[138].

$$\mathrm{RMSD} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - x'_i)^2 + (y_i - y'_i)^2 + (z_i - z'_i)^2} \qquad (1.1)$$

where x_i, y_i, z_i are the coordinates of the *i*th atom of the docked ligand, and x'_i, y'_i, z'_i are those of the complexed ligand in its native binding pose. SR can also be defined in different ways depending on which docked pose is compared to the true pose—most widely used choices are looking at top 1, top 3, or top 5 poses (**Eq. 1.2**). Typically, the RMSD value cutoff for a pose to be considered "good" (near-native pose) is 2 Å[139].

$$\mathrm{SR_{TopN}} = \frac{\#\text{ of systems with a good pose ranked in top N}}{\text{total \# of docked systems}} \times 100\ (\%) \qquad (1.2)$$

Another commonly used metric for docking power is receiver operating curve (ROC), which measures the ability of the chosen method to differentiate between correct and incorrect binding poses. ROC plots sensitivity (proportion of correctly classified positive samples) vs. specificity (proportion of correctly classified negative samples) at different thresholds of the docking score, or in the case of ML classifiers, predicted probabilities. ROC curves are quantified by computing the area under the curve (AUC), where an AUC of 0.5 indicates a method performing no better than random guessing, and an AUC of 1 indicates perfect discrimination between good and bad poses[71].

1.3.2.2 Community Benchmarks for Scoring

The following section outlines community benchmarking sets that are available for the retrospective evaluation of SFs, both classical and ML-based. Most of the benchmarks are designed to evaluate a SF's ability to correctly predict the binding affinity (scoring) or identify potent molecules (screening). Since there are not many benchmarks specifically designed for docking power, benchmarks for scoring in lieu of docking are discussed below.

Comparative assessment of scoring functions (CASF) is perhaps the most widely used benchmark that contains protein-ligand structures binding affinities, and thus used for evaluating scoring, ranking, and docking powers. CASF-2007, the first of its versions, is based on PDBbind v.2007 and contains 195 diverse protein-ligand complexes with high-resolution structures and known binding constants[140]. Its successor, CASF-2013, obtains the protein-ligand structures from an updated PDBbind v.2013 refine set, and incorporates protein sequence clustering to further reduce bias from similar molecular structures[136,141]. The latest version, CASF-2016, is updated based on PDBbind v.2016 core set and includes 285 high-resolution structures, as well as a decoy set consisting of cross-docked ligands[139]. Community structure activity resource (CSAR)[142] and its later releases are another scoring dataset containing diverse protein-ligand complexes from PDBbind, augmented with binding affinity data from BindingMOAD[143]. In addition to screening- or scoring-specific benchmarks, recent acceleration of ML-based algorithms in the field of biomolecular sciences has sparked a new interest in developing large-scale datasets and evaluation tools for molecular ML tasks, including QSAR, ADMET predictions, binding constant predictions, and quantum chemistry calculations. Three such datasets, DeepChem's MoleculeNet[144], therapeutic data commons (TDC)[145], and most recently, WelQrate[146], all provide solid screening and scoring benchmarks for MLSFs. Despite decades of efforts to create unbiased community benchmarks, the same challenges persist, continuing to hinder those who seek to evaluate and compare both established and emerging SFs, whether ML-based or not[91,147,148]. Benchmarking limitations specific to (and prominent in) MLSFs and strategies to address some of these drawbacks will be discussed in more detail in Section 1.3.3.

1.3.2.3 Community Benchmarks for Screening

Directory of useful decoys (DUD) is one of the first publicly available benchmarking

sets for screening and contains 40 diverse targets[149]. Previous studies have shown that selecting completely random molecules as decoys led to a common issue of artificial enrichment, as the validation measures the ability of an SF to differentiate between binders and nonbinders based on simple 1D properties of molecules[150]. Therefore, the focus of the VS benchmarking sets, such as DUD, is to create a decoy set that is physically similar to the corresponding active. For example, DUD's successor, directory of useful decoys-enhanced (DUD-E), improved its decoy set by matching the property of formal charge and excluding decoys that turned out to be binders in DUD[137]. Another widespread issue in previous benchmarks that DUD-E attempted to address was analogue bias, which arises from the over-representation of certain chemotypes that dominate and bias docking results, leading to an overestimation of VS performance[151]. The authors of DUD-E diversified ligands in the dataset based on a structural clustering method such as Bemis-Murcko scaffold[137]. Similar strategies are adopted for other benchmarking sets such as maximum unbiased validation (MUV), which is designed based on PubChem bioactivity data and implements nearest neighbor analysis to select actives that are (1) well-embedded in decoys and (2) certain distances apart from one another in the chemical space[152]. Demanding evaluation kits for objective *in silico* screening, or DEKOIS, is another VS benchmarking set that allows users to evaluate their methods on a target of choice by providing an algorithm to create a set of tailor-made decoys for any actives[153]. Given that the results of docking can be target-dependent[154], this is a particularly useful strategy for users who want to select an SF that best suits their needs in a VS campaign. There is also DEKOIS 2.0. which aims to improve speed, active-decoy property matching, and diversity of molecules in the previous version[155]. In recent years, the development of new and better benchmarks to reduce bias has continued, with such datasets as DUDE-Z, which is an optimized version of DUD-E that aims to remove charge-matching after protonation in 3D conformation[156], and LIT-PCBA, which is

particularly suited for unbiasing MLSFs[157].

1.3.3 Pitfalls of MLSFs

While these AI-powered models show a promising future for the prediction of protein-ligand interaction to accelerate early-stage drug discovery, it is crucial to be aware of common pitfalls in ML. Numerous SFs are introduced each year claiming superior performance, but many of those studies are evaluated on flawed benchmarks that have been repeatedly criticized. Recent studies rightly pointed out and gave evidence for hidden biases in validation methods, leading to overoptimistic results using MLSFs. Sieg et al. demonstrated that the DUD and DUD-E benchmarks—originally designed for conventional SFs—contain inherent biases, such as unmatched 2D topological features between actives and decoys[158], which the authors acknowledged as a critical flaw in the datasets[137]. These biases can be readily exploited by MLSFs using simple chemical descriptors, leading to over-optimistic performances. This study concluded that MLSFs trained and validated on such datasets learned to discriminate active ligands against inactives based on low-dimensional ligand features rather than discerning the interactions that truly influence the binding between the protein and ligand[158]. Further, two studies demonstrated that state-of-the-art GNNs trained for binding affinity predictions are prone to ligand memorization rather than learning physicochemical interactions required to generalize to new systems[159,160]. The prominence of analogue bias and artificial enrichment in MLSFs is not surprising given that the training and validation data often come from the same source. For example, SFs for scoring are often validated on a subdivision called the “core” subset of PDBbind dataset while being trained on the rest of the data belonging to “general” and/or “refined” subsets in CASF-2016 benchmark[139]. For SBVS, active-decoy datasets such as DUD and DEKOIS are usually split into training and test sets with some debiasing techniques to prevent data overlap. However, a

recent study by Li et al. revealed that such common splitting methods are not enough to prevent data leakage due to the presence of highly similar proteins or ligands that occur in both the training and test sets[161]. They showed that careful partitioning of training, validation, and test sets based on ligand fingerprint and protein sequence similarity resulted in an improved generalization of MLSFs when the same dataset was retrained on this new split[161]. The final pitfall of applying ML in this field—one particularly relevant to MLSFs—becomes evident when evaluating the performance of the same SF across different tasks. Several studies have already shown that many MLSFs trained for one task perform poorly in other tasks, demonstrating the data-sensitivity of these models[87,90]. For example, MLSF trained for BAP will perform poorly in SBVS since the majority of the molecules in the library are inactives, are not considered in the training. Even for SFs trained for SBVS tasks, the ratio of decoys to actives in the training set is significantly overestimated in comparison to real datasets which consist of a far larger proportion of inactive molecules[70]. While prospective validation in experimental settings remains the ultimate test of a SF's generalizability and effectiveness, retrospective benchmarking is an essential tool for both users and developers to assess the performance of MLSFs—whether for applications or development.

1.4 Project Motivation and Goals

As briefly discussed in Section 1.2.2, computational methods hold great advantages in small-molecule ligand discovery for challenging targets such as PPIs. As researchers increasingly focus on exploring new PPI targets and elucidating their structures, the application of ML techniques to the discovery of PPI inhibitors is expected to be particularly fruitful. However, despite the availability of growing structural databases and the increasing number of newly published MLSFs each year, many still exhibit poor generalization to unseen targets during training, highlighting the need for target-specific models readily

applicable for real-world VS scenarios[160,162–166]. In fact, developing target-specific models have become a frequent strategy to improve performance of MLSFs for VS, rather than training on a large and diverse dataset with limited applicability[85,114,100,167].

While docking campaigns have been carried out targeting PPI pockets with MLSFs before[65–67], there are currently no SFs specifically designed for PPI targets, nor a database curated for training and validating ML-based tools for PPI inhibitors. Given the distinctive characteristics of PPI interfaces that separate them from conventional targets, MLSFs are expected to struggle even more in generalizing to PPI targets. To address this challenge, we set the following goals of the thesis:

- Construct 3D structural databases for the training and validation of docking power for PPI targets by generating poses of known PPI inhibitors docked into their pockets.
- Develop target-specific MLSFs for the prediction of accurate binding poses to improve molecular docking against PPI targets.

In summary, this thesis seeks to construct a database of docked poses for PPI inhibitors to facilitate structure-based PPI modulator discovery, and to design domain-specific methods that surpass current state-of-the-art MLSFs in the prediction of accurate binding modes of PPI inhibitors.

1.5 References

(1) Dowden, H.; Munro, J. Trends in Clinical Success Rates and Therapeutic Focus. *Nature Reviews Drug Discovery*. 18th ed. 2019, pp 495–497.

(2) Austin, D.; Hayford, T. Research and Development in the Pharmaceutical Industry. *CBO*. 2021. https://www.cbo.gov/publication/57126#_idTextAnchor000.

(3) Sun, D.; Gao, W.; Hu, H.; Zhou, S. Why 90% of Clinical Drug Development Fails and How to Improve It? *Acta Pharm. Sin. B* **2022**, *12* (7), 3049–3062. https://doi.org/10.1016/j.apsb.2022.02.002.

(4) Mater, A. C.; Coote, M. L. Deep Learning in Chemistry. *J. Chem. Inf. Model.* **2019**, *59* (6), 2545–2559. https://doi.org/10.1021/acs.jcim.9b00266.

(5) Bhardwaj, A.; Kishore, S.; Pandey, D. K. Artificial Intelligence in Biological Sciences. *Life* **2022**, *12* (9), 1430. https://doi.org/10.3390/life12091430.

(6) Paul, D.; Sanap, G.; Shenoy, S.; Kalyane, D.; Kalia, K.; Tekade, R. K. Artificial Intelligence in Drug Discovery and Development. *Drug Discov. Today* **2021**, *26* (1), 80–93. https://doi.org/10.1016/j.drudis.2020.10.010.

(7) Vamathevan, J.; Clark, D.; Czodrowski, P.; Dunham, I.; Ferran, E.; Lee, G.; Li, B.; Madabhushi, A.; Shah, P.; Spitzer, M.; Zhao, S. Applications of Machine Learning in Drug Discovery and Development. *Nat. Rev. Drug Discov.* **2019**, *18* (6), 463–477. https://doi.org/10.1038/s41573-019-0024-5.

(8) Burley, S. K.; Bhatt, R.; Bhikadiya, C.; Bi, C.; Biester, A.; Biswas, P.; Bittrich, S.; Blaumann, S.; Brown, R.; Chao, H.; Chithari, V. R.; Craig, P. A.; Crichlow, G. V.; Duarte, J. M.; Dutta, S.; Feng, Z.; Flatt, J. W.; Ghosh, S.; Goodsell, D. S.; Green, R. K.; Guranovic, V.; Henry, J.; Hudson, B. P.; Joy, M.; Kaelber, J. T.; Khokhriakov, I.; Lai, J.-S.; Lawson, C. L.; Liang, Y.; Myers-Turnbull, D.; Peisach, E.; Persikova, I.; Piehl, D. W.; Pingale, A.; Rose, Y.; Sagendorf, J.; Sali, A.; Segura, J.; Sekharan, M.; Shao, C.; Smith, J.; Trumbull, M.; Vallat, B.; Voigt, M.; Webb, B.; Whetstone, S.; Wu-Wu, A.; Xing, T.; Young, J. Y.; Zalevsky, A.; Zardecki, C. Updated Resources for Exploring Experimentally-Determined PDB Structures and Computed Structure Models at the RCSB Protein Data Bank. *Nucleic Acids Res.* **2025**, *53* (D1), D564–D574. https://doi.org/10.1093/nar/gkae1091.

(9) Grygorenko, O. O.; Radchenko, D. S.; Dziuba, I.; Chuprina, A.; Gubina, K. E.; Moroz, Y. S. Generating Multibillion Chemical Space of Readily Accessible Screening Compounds. *iScience* **2020**, *23* (11), 101681. https://doi.org/10.1016/j.isci.2020.101681.

(10) Sadybekov, A. V.; Katritch, V. Computational Approaches Streamlining Drug Discovery. *Nature* **2023**, *616* (7958), 673–685. https://doi.org/10.1038/s41586-023-05905-z.

(11) Pandey, M.; Fernandez, M.; Gentile, F.; Isayev, O.; Tropsha, A.; Stern, A. C.; Cherkasov, A. The Transformational Role of GPU Computing and Deep Learning in Drug Discovery. *Nat. Mach. Intell.* **2022**, *4* (3), 211–221. https://doi.org/10.1038/s42256-022-00463-x.

(12) Xia, S.; Chen, E.; Zhang, Y. Integrated Molecular Modeling and Machine Learning for Drug Design. *J. Chem. Theory Comput.* **2023**, *19* (21), 7478–7495. https://doi.org/10.1021/acs.jctc.3c00814.

(13) Dugger, S. A.; Platt, A.; Goldstein, D. B. Drug Development in the Era of Precision Medicine. *Nat. Rev. Drug Discov.* **2018**, *17* (3), 183–196. https://doi.org/10.1038/nrd.2017.226.

(14) You, Y.; Lai, X.; Pan, Y.; Zheng, H.; Vera, J.; Liu, S.; Deng, S.; Zhang, L. Artificial Intelligence in Cancer Target Identification and Drug Discovery. *Signal Transduct. Target. Ther.* **2022**, *7* (1), 156. https://doi.org/10.1038/s41392-022-00994-0.
(15) Himmelstein, D. S.; Baranzini, S. E. Heterogeneous Network Edge Prediction: A Data Integration Approach to Prioritize Disease-Associated Genes. *PLOS Comput. Biol.* **2015**, *11* (7), e1004259. https://doi.org/10.1371/journal.pcbi.1004259.
(16) Hasselgren, C.; Oprea, T. I. Artificial Intelligence for Drug Discovery: Are We There Yet? *Annu. Rev. Pharmacol. Toxicol.* **2024**, *64* (1), 527–550. https://doi.org/10.1146/annurev-pharmtox-040323-040828.
(17) Han, R.; Yoon, H.; Kim, G.; Lee, H.; Lee, Y. Revolutionizing Medicinal Chemistry: The Application of Artificial Intelligence (AI) in Early Drug Discovery. *Pharmaceuticals* **2023**, *16* (9), 1259. https://doi.org/10.3390/ph16091259.
(18) Senior, A. W.; Evans, R.; Jumper, J.; Kirkpatrick, J.; Sifre, L.; Green, T.; Qin, C.; Žídek, A.; Nelson, A. W. R.; Bridgland, A.; Penedones, H.; Petersen, S.; Simonyan, K.; Crossan, S.; Kohli, P.; Jones, D. T.; Silver, D.; Kavukcuoglu, K.; Hassabis, D. Improved Protein Structure Prediction Using Potentials from Deep Learning. *Nature* **2020**, *577* (7792), 706–710. https://doi.org/10.1038/s41586-019-1923-7.
(19) Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Žídek, A.; Potapenko, A.; Bridgland, A.; Meyer, C.; Kohl, S. A. A.; Ballard, A. J.; Cowie, A.; Romera-Paredes, B.; Nikolov, S.; Jain, R.; Adler, J.; Back, T.; Petersen, S.; Reiman, D.; Clancy, E.; Zielinski, M.; Steinegger, M.; Pacholska, M.; Berghammer, T.; Bodenstein, S.; Silver, D.; Vinyals, O.; Senior, A. W.; Kavukcuoglu, K.; Kohli, P.; Hassabis, D. Highly Accurate Protein Structure Prediction with AlphaFold. *Nature* **2021**, *596* (7873), 583–589. https://doi.org/10.1038/s41586-021-03819-2.
(20) Abramson, J.; Adler, J.; Dunger, J.; Evans, R.; Green, T.; Pritzel, A.; Ronneberger, O.; Willmore, L.; Ballard, A. J.; Bambrick, J.; Bodenstein, S. W.; Evans, D. A.; Hung, C.-C.; O'Neill, M.; Reiman, D.; Tunyasuvunakool, K.; Wu, Z.; Žemgulytė, A.; Arvaniti, E.; Beattie, C.; Bertolli, O.; Bridgland, A.; Cherepanov, A.; Congreve, M.; Cowen-Rivers, A. I.; Cowie, A.; Figurnov, M.; Fuchs, F. B.; Gladman, H.; Jain, R.; Khan, Y. A.; Low, C. M. R.; Perlin, K.; Potapenko, A.; Savy, P.; Singh, S.; Stecula, A.; Thillaisundaram, A.; Tong, C.; Yakneen, S.; Zhong, E. D.; Zielinski, M.; Žídek, A.; Bapst, V.; Kohli, P.; Jaderberg, M.; Hassabis, D.; Jumper, J. M. Accurate Structure Prediction of Biomolecular Interactions with AlphaFold 3. *Nature* **2024**, *630* (8016), 493–500. https://doi.org/10.1038/s41586-024-07487-w.
(21) Evans, R.; O'Neill, M.; Pritzel, A.; Antropova, N.; Senior, A.; Green, T.; Žídek, A.; Bates, R.; Blackwell, S.; Yim, J.; Ronneberger, O.; Bodenstein, S.; Zielinski, M.; Bridgland, A.; Potapenko, A.; Cowie, A.; Tunyasuvunakool, K.; Jain, R.; Clancy, E.; Kohli, P.; Jumper, J.; Hassabis, D. Protein Complex Prediction with AlphaFold-Multimer. October 4, 2021. https://doi.org/10.1101/2021.10.04.463034.
(22) Gao, M.; Nakajima An, D.; Parks, J. M.; Skolnick, J. AF2Complex Predicts Direct Physical Interactions in Multimeric Proteins with Deep Learning. *Nat. Commun.* **2022**, *13* (1), 1744. https://doi.org/10.1038/s41467-022-29394-2.
(23) Lin, Z.; Akin, H.; Rao, R.; Hie, B.; Zhu, Z.; Lu, W.; Smetanin, N.; Verkuil, R.; Kabeli, O.; Shmueli, Y.; Dos Santos Costa, A.; Fazel-Zarandi, M.; Sercu, T.; Candido, S.; Rives, A. Evolutionary-Scale Prediction of Atomic-Level Protein Structure with a Language Model. *Science* **2023**, *379* (6637), 1123–1130. https://doi.org/10.1126/science.ade2574.
(24) Wohlwend, J.; Corso, G.; Passaro, S.; Getz, N.; Reveiz, M.; Leidal, K.; Swiderski, W.;

Atkinson, L.; Portnoi, T.; Chinn, I.; Silterra, J.; Jaakkola, T.; Barzilay, R. Boltz-1 Democratizing Biomolecular Interaction Modeling. November 20, 2024. https://doi.org/10.1101/2024.11.19.624167.

(25) Kandel, J.; Tayara, H.; Chong, K. T. PUResNet: Prediction of Protein-Ligand Binding Sites Using Deep Residual Neural Network. *J. Cheminformatics* **2021**, *13* (1), 65. https://doi.org/10.1186/s13321-021-00547-7.

(26) Desaphy, J.; Bret, G.; Rognan, D.; Kellenberger, E. Sc-PDB: A 3D-Database of Ligandable Binding Sites—10 Years On. *Nucleic Acids Res.* **2015**, *43* (D1), D399–D404. https://doi.org/10.1093/nar/gku928.

(27) Jiménez, J.; Doerr, S.; Martínez-Rosell, G.; Rose, A. S.; De Fabritiis, G. DeepSite: Protein-Binding Site Predictor Using 3D-Convolutional Neural Networks. *Bioinformatics* **2017**, *33* (19), 3036–3042. https://doi.org/10.1093/bioinformatics/btx350.

(28) Meller, A.; Ward, M.; Borowsky, J.; Kshirsagar, M.; Lotthammer, J. M.; Oviedo, F.; Ferres, J. L.; Bowman, G. R. Predicting Locations of Cryptic Pockets from Single Protein Structures Using the PocketMiner Graph Neural Network. *Nat. Commun.* **2023**, *14* (1), 1177. https://doi.org/10.1038/s41467-023-36699-3.

(29) Vemula, D.; Jayasurya, P.; Sushmitha, V.; Kumar, Y. N.; Bhandari, V. CADD, AI and ML in Drug Discovery: A Comprehensive Review. *Eur. J. Pharm. Sci.* **2023**, *181*, 106324. https://doi.org/10.1016/j.ejps.2022.106324.

(30) Muratov, E. N.; Bajorath, J.; Sheridan, R. P.; Tetko, I. V.; Filimonov, D.; Poroikov, V.; Oprea, T. I.; Baskin, I. I.; Varnek, A.; Roitberg, A.; Isayev, O.; Curtalolo, S.; Fourches, D.; Cohen, Y.; Aspuru-Guzik, A.; Winkler, D. A.; Agrafiotis, D.; Cherkasov, A.; Tropsha, A. QSAR without Borders. *Chem. Soc. Rev.* **2020**, *49* (11), 3525–3564. https://doi.org/10.1039/D0CS00098A.

(31) Priya, S.; Tripathi, G.; Singh, D. B.; Jain, P.; Kumar, A. Machine Learning Approaches and Their Applications in Drug Discovery and Design. *Chem. Biol. Drug Des.* **2022**, *100* (1), 136–153. https://doi.org/10.1111/cbdd.14057.

(32) Ma, J.; Sheridan, R. P.; Liaw, A.; Dahl, G. E.; Svetnik, V. Deep Neural Nets as a Method for Quantitative Structure–Activity Relationships. *J. Chem. Inf. Model.* **2015**, *55* (2), 263–274. https://doi.org/10.1021/ci500747n.

(33) Kwon, S.; Bae, H.; Jo, J.; Yoon, S. Comprehensive Ensemble in QSAR Prediction for Drug Discovery. *BMC Bioinformatics* **2019**, *20* (1), 521. https://doi.org/10.1186/s12859-019-3135-4.

(34) Cheng, T.; Li, Q.; Zhou, Z.; Wang, Y.; Bryant, S. H. Structure-Based Virtual Screening for Drug Discovery: A Problem-Centric Review. *AAPS J.* **2012**, *14* (1), 133–141. https://doi.org/10.1208/s12248-012-9322-0.

(35) Varela-Rial, A.; Majewski, M.; De Fabritiis, G. Structure Based Virtual Screening: Fast and Slow. *WIREs Comput. Mol. Sci.* **2022**, *12* (2), e1544. https://doi.org/10.1002/wcms.1544.

(36) Cleves, A. E.; Jain, A. N. Effects of Inductive Bias on Computational Evaluations of Ligand-Based Modeling and on Drug Discovery. *J. Comput. Aided Mol. Des.* **2008**, *22* (3–4), 147–159. https://doi.org/10.1007/s10822-007-9150-y.

(37) Thomas, M.; Smith, R. T.; O'Boyle, N. M.; De Graaf, C.; Bender, A. Comparison of Structure- and Ligand-Based Scoring Functions for Deep Generative Models: A GPCR Case Study. *J. Cheminformatics* **2021**, *13* (1), 39. https://doi.org/10.1186/s13321-021-00516-0.

(38) Paggi, J. M.; Pandit, A.; Dror, R. O. The Art and Science of Molecular Docking. *Annu. Rev. Biochem.* **2024**, *93* (1), 389–410. https://doi.org/10.1146/annurev-biochem-

030222-120000.
(39) Gorgulla, C. Recent Developments in Ultralarge and Structure-Based Virtual Screening Approaches. *Annu. Rev. Biomed. Data Sci.* **2023**, *6* (1), 229–258. https://doi.org/10.1146/annurev-biodatasci-020222-025013.
(40) Pantsar, T.; Poso, A. Binding Affinity via Docking: Fact and Fiction. *Molecules* **2018**, *23* (8), 1899. https://doi.org/10.3390/molecules23081899.
(41) Kuan, J.; Radaeva, M.; Avenido, A.; Cherkasov, A.; Gentile, F. Keeping Pace with the Explosive Growth of Chemical Libraries with Structure-based Virtual Screening. *WIREs Comput. Mol. Sci.* **2023**, *13* (6), e1678. https://doi.org/10.1002/wcms.1678.
(42) Cavasotto, C. N.; Di Filippo, J. I. The Impact of Supervised Learning Methods in Ultralarge High-Throughput Docking. *J. Chem. Inf. Model.* **2023**, *63* (8), 2267–2280. https://doi.org/10.1021/acs.jcim.2c01471.
(43) Berenger, F.; Kumar, A.; Zhang, K. Y. J.; Yamanishi, Y. Lean-Docking: Exploiting Ligands' Predicted Docking Scores to Accelerate Molecular Docking. *J. Chem. Inf. Model.* **2021**, *61* (5), 2341–2352. https://doi.org/10.1021/acs.jcim.0c01452.
(44) Gentile, F.; Agrawal, V.; Hsing, M.; Ton, A.-T.; Ban, F.; Norinder, U.; Gleave, M. E.; Cherkasov, A. Deep Docking: A Deep Learning Platform for Augmentation of Structure Based Drug Discovery. *ACS Cent. Sci.* **2020**, *6* (6), 939–949. https://doi.org/10.1021/acscentsci.0c00229.
(45) Gentile, F.; Fernandez, M.; Ban, F.; Ton, A.-T.; Mslati, H.; Perez, C. F.; Leblanc, E.; Yaacoub, J. C.; Gleave, J.; Stern, A.; Wong, B.; Jean, F.; Strynadka, N.; Cherkasov, A. Automated Discovery of Noncovalent Inhibitors of SARS-CoV-2 Main Protease by Consensus Deep Docking of 40 Billion Small Molecules. *Chem. Sci.* **2021**, *12* (48), 15960–15974. https://doi.org/10.1039/D1SC05579H.
(46) Gu, S.; Shen, C.; Zhang, X.; Sun, H.; Cai, H.; Luo, H.; Zhao, H.; Liu, B.; Du, H.; Zhao, Y.; Fu, C.; Zhai, S.; Deng, Y.; Liu, H.; Hou, T.; Kang, Y. Benchmarking AI-Powered Docking Methods from the Perspective of Virtual Screening. *Nat. Mach. Intell.* **2025**, *7* (3), 509–520. https://doi.org/10.1038/s42256-025-00993-0.
(47) Mabonga, L.; Kappo, A. P. Protein-Protein Interaction Modulators: Advances, Successes and Remaining Challenges. *Biophys. Rev.* **2019**, *11* (4), 559–581. https://doi.org/10.1007/s12551-019-00570-x.
(48) *Recent advances in the development of protein–protein interactions modulators: mechanisms and clinical trials | Signal Transduction and Targeted Therapy*. https://www.nature.com/articles/s41392-020-00315-3 (accessed 2024-05-21).
(49) Xie, X.; Yu, T.; Li, X.; Zhang, N.; Foster, L. J.; Peng, C.; Huang, W.; He, G. Recent Advances in Targeting the "Undruggable" Proteins: From Drug Discovery to Clinical Trials. *Signal Transduct. Target. Ther.* **2023**, *8* (1), 335. https://doi.org/10.1038/s41392-023-01589-z.
(50) Milroy, L.-G.; Grossmann, T. N.; Hennig, S.; Brunsveld, L.; Ottmann, C. Modulators of Protein–Protein Interactions. *Chem. Rev.* **2014**, *114* (9), 4695–4748. https://doi.org/10.1021/cr400698c.
(51) Ran, X.; Gestwicki, J. E. Inhibitors of Protein–Protein Interactions (PPIs): An Analysis of Scaffold Choices and Buried Surface Area. *Curr. Opin. Chem. Biol.* **2018**, *44*, 75–86. https://doi.org/10.1016/j.cbpa.2018.06.004.
(52) Cesa, L. C.; Mapp, A. K.; Gestwicki, J. E. Direct and Propagated Effects of Small Molecules on Protein–Protein Interaction Networks. *Front. Bioeng. Biotechnol.* **2015**, *3*. https://doi.org/10.3389/fbioe.2015.00119.
(53) Robertson, N. S.; Spring, D. R. Using Peptidomimetics and Constrained Peptides as Valuable Tools for Inhibiting Protein–Protein Interactions. *Molecules* **2018**, *23* (4),

959. https://doi.org/10.3390/molecules23040959.
(54) Cheng, S.-S.; Yang, G.-J.; Wang, W.; Leung, C.-H.; Ma, D.-L. The Design and Development of Covalent Protein-Protein Interaction Inhibitors for Cancer Treatment. *J. Hematol. Oncol.J Hematol Oncol* **2020**, *13* (1), 26. https://doi.org/10.1186/s13045-020-00850-0.
(55) Ivanov, A. A.; Khuri, F. R.; Fu, H. Targeting Protein–Protein Interactions as an Anticancer Strategy. *Trends Pharmacol. Sci.* **2013**, *34* (7), 393–400. https://doi.org/10.1016/j.tips.2013.04.007.
(56) Smith, M. C.; Gestwicki, J. E. Features of Protein–Protein Interactions That Translate into Potent Inhibitors: Topology, Surface Area and Affinity. *Expert Rev. Mol. Med.* **2012**, *14*, e16. https://doi.org/10.1017/erm.2012.10.
(57) Kuenemann, M. A.; Bourbon, L. M. L.; Labbé, C. M.; Villoutreix, B. O.; Sperandio, O. Which Three-Dimensional Characteristics Make Efficient Inhibitors of Protein–Protein Interactions? *J. Chem. Inf. Model.* **2014**, *54* (11), 3067–3079. https://doi.org/10.1021/ci500487q.
(58) Buchwald, P. Small-molecule Protein–Protein Interaction Inhibitors: Therapeutic Potential in Light of Molecular Size, Chemical Space, and Ligand Binding Efficiency Considerations. *IUBMB Life* **2010**, *62* (10), 724–731. https://doi.org/10.1002/iub.383.
(59) Cukuroglu, E.; Engin, H. B.; Gursoy, A.; Keskin, O. Hot Spots in Protein–Protein Interfaces: Towards Drug Discovery. *Prog. Biophys. Mol. Biol.* **2014**, *116* (2–3), 165–173. https://doi.org/10.1016/j.pbiomolbio.2014.06.003.
(60) Magee, T. V. Progress in Discovery of Small-Molecule Modulators of Protein–Protein Interactions via Fragment Screening. *Bioorg. Med. Chem. Lett.* **2015**, *25* (12), 2461–2468. https://doi.org/10.1016/j.bmcl.2015.04.089.
(61) Robson-Tull, J. Biophysical Screening in Fragment-Based Drug Design: A Brief Overview. *Biosci. Horiz. Int. J. Stud. Res.* **2018**, *11*. https://doi.org/10.1093/biohorizons/hzy015.
(62) Erlanson, D. A.; Fesik, S. W.; Hubbard, R. E.; Jahnke, W.; Jhoti, H. Twenty Years on: The Impact of Fragments on Drug Discovery. *Nat. Rev. Drug Discov.* **2016**, *15* (9), 605–619. https://doi.org/10.1038/nrd.2016.109.
(63) Frackenpohl, J.; Arvidsson, P. I.; Schreiber, J. V.; Seebach, D. The Outstanding Biological Stability Ofβ- Andγ-Peptides toward Proteolytic Enzymes: An In Vitro Investigation with Fifteen Peptidases. *ChemBioChem* **2001**, *2* (6), 445–455. https://doi.org/10.1002/1439-7633(20010601)2:6<445::AID-CBIC445>3.0.CO;2-R.
(64) Hall, D. R.; Kozakov, D.; Vajda, S. Analysis of Protein Binding Sites by Computational Solvent Mapping. In *Computational Drug Discovery and Design*; Baron, R., Ed.; Methods in Molecular Biology; Springer New York: New York, NY, 2012; Vol. 819, pp 13–27. https://doi.org/10.1007/978-1-61779-465-0_2.
(65) Scheper, J.; Guerra-Rebollo, M.; Sanclimens, G.; Moure, A.; Masip, I.; González-Ruiz, D.; Rubio, N.; Crosas, B.; Meca-Cortés, Ó.; Loukili, N.; Plans, V.; Morreale, A.; Blanco, J.; Ortiz, A. R.; Messeguer, À.; Thomson, T. M. Protein-Protein Interaction Antagonists as Novel Inhibitors of Non-Canonical Polyubiquitylation. *PLoS ONE* **2010**, *5* (6), e11403. https://doi.org/10.1371/journal.pone.0011403.
(66) Lawrence, H. R.; Li, Z.; Richard Yip, M. L.; Sung, S.-S.; Lawrence, N. J.; McLaughlin, M. L.; McManus, G. J.; Zaworotko, M. J.; Sebti, S. M.; Chen, J.; Guida, W. C. Identification of a Disruptor of the MDM2-P53 Protein–Protein Interaction Facilitated by High-Throughput in Silico Docking. *Bioorg. Med. Chem. Lett.* **2009**, *19* (14), 3756–3759. https://doi.org/10.1016/j.bmcl.2009.04.124.
(67) Tian, W.; Han, X.; Yan, M.; Xu, Y.; Duggineni, S.; Lin, N.; Luo, G.; Li, Y. M.; Han, X.;

Huang, Z.; An, J. Structure-Based Discovery of a Novel Inhibitor Targeting the β-Catenin/Tcf4 Interaction. *Biochemistry* **2012**, *51* (2), 724–731. https://doi.org/10.1021/bi201428h.
(68) Pagadala, N. S.; Syed, K.; Tuszynski, J. Software for Molecular Docking: A Review. *Biophys. Rev.* **2017**, *9* (2), 91–102. https://doi.org/10.1007/s12551-016-0247-1.
(69) Kuntz, I. D.; Blaney, J. M.; Oatley, S. J.; Langridge, R.; Ferrin, T. E. A Geometric Approach to Macromolecule-Ligand Interactions. *J. Mol. Biol.* **1982**, *161* (2), 269–288. https://doi.org/10.1016/0022-2836(82)90153-X.
(70) Li, H.; Sze, K.-H.; Lu, G.; Ballester, P. J. Machine-Learning Scoring Functions for Structure-Based Virtual Screening. *WIREs Comput. Mol. Sci.* **2021**, *11* (1), e1478. https://doi.org/10.1002/wcms.1478.
(71) *Machine-learning scoring functions to improve structure-based binding affinity prediction and virtual screening - Ain - 2015 - WIREs Computational Molecular Science - Wiley Online Library*. https://wires.onlinelibrary.wiley.com/doi/full/10.1002/wcms.1225 (accessed 2023-12-30).
(72) Shen, C.; Ding, J.; Wang, Z.; Cao, D.; Ding, X.; Hou, T. From Machine Learning to Deep Learning: Advances in Scoring Functions for Protein–Ligand Docking. *WIREs Comput. Mol. Sci.* **2020**, *10* (1), e1429. https://doi.org/10.1002/wcms.1429.
(73) Li, J.; Fu, A.; Zhang, L. An Overview of Scoring Functions Used for Protein–Ligand Interactions in Molecular Docking. *Interdiscip. Sci. Comput. Life Sci.* **2019**, *11* (2), 320–328. https://doi.org/10.1007/s12539-019-00327-w.
(74) Muegge, I.; Martin, Y. C. A General and Fast Scoring Function for Protein−Ligand Interactions: A Simplified Potential Approach. *J. Med. Chem.* **1999**, *42* (5), 791–804. https://doi.org/10.1021/jm980536j.
(75) Baxter, C. A.; Murray, C. W.; Clark, D. E.; Westhead, D. R.; Eldridge, M. D. Flexible Docking Using Tabu Search and an Empirical Estimate of Binding Affinity. *Proteins Struct. Funct. Genet.* **1998**, *33* (3), 367–382. https://doi.org/10.1002/(SICI)1097-0134(19981115)33:3<367::AID-PROT6>3.0.CO;2-W.
(76) Velec, H. F. G.; Gohlke, H.; Klebe, G. DrugScore^{CSD} Knowledge-Based Scoring Function Derived from Small Molecule Crystal Data with Superior Recognition Rate of Near-Native Ligand Poses and Better Affinity Prediction. *J. Med. Chem.* **2005**, *48* (20), 6296–6303. https://doi.org/10.1021/jm050436v.
(77) Wang, R.; Lai, L.; Wang, S. Further Development and Validation of Empirical Scoring Functions for Structure-Based Binding Affinity Prediction. *J. Comput. Aided Mol. Des.* **2002**, *16* (1), 11–26. https://doi.org/10.1023/A:1016357811882.
(78) The Sybyl Software, 2006.
(79) Friesner, R. A.; Murphy, R. B.; Repasky, M. P.; Frye, L. L.; Greenwood, J. R.; Halgren, T. A.; Sanschagrin, P. C.; Mainz, D. T. Extra Precision Glide: Docking and Scoring Incorporating a Model of Hydrophobic Enclosure for Protein−Ligand Complexes. *J. Med. Chem.* **2006**, *49* (21), 6177–6196. https://doi.org/10.1021/jm051256o.
(80) Trott, O.; Olson, A. J. AutoDock Vina: Improving the Speed and Accuracy of Docking with a New Scoring Function, Efficient Optimization, and Multithreading. *J. Comput. Chem.* **2010**, *31* (2), 455–461. https://doi.org/10.1002/jcc.21334.
(81) Nguyen, D. D.; Wei, G.-W. AGL-Score: Algebraic Graph Learning Score for Protein–Ligand Binding Scoring, Ranking, Docking, and Screening. *J. Chem. Inf. Model.* **2019**, *59* (7), 3291–3304. https://doi.org/10.1021/acs.jcim.9b00334.
(82) Sun, H.; Pan, P.; Tian, S.; Xu, L.; Kong, X.; Li, Y.; Dan Li; Hou, T. Constructing and Validating High-Performance MIEC-SVM Models in Virtual Screening for Kinases: A

Better Way for Actives Discovery. *Sci. Rep.* **2016**, *6* (1), 24817. https://doi.org/10.1038/srep24817.
(83) Ballester, P. J.; Mitchell, J. B. O. A Machine Learning Approach to Predicting Protein–Ligand Binding Affinity with Applications to Molecular Docking. *Bioinformatics* **2010**, *26* (9), 1169–1175. https://doi.org/10.1093/bioinformatics/btq112.
(84) Li, H.; Leung, K.; Wong, M.; Ballester, P. J. Improving AutoDock Vina Using Random Forest: The Growing Accuracy of Binding Affinity Prediction by the Effective Exploitation of Larger Data Sets. *Mol. Inform.* **2015**, *34* (2–3), 115–126. https://doi.org/10.1002/minf.201400132.
(85) Guedes, I. A.; Barreto, A. M. S.; Marinho, D.; Krempser, E.; Kuenemann, M. A.; Sperandio, O.; Dardenne, L. E.; Miteva, M. A. New Machine Learning and Physics-Based Scoring Functions for Drug Discovery. *Sci. Rep.* **2021**, *11* (1), 3198. https://doi.org/10.1038/s41598-021-82410-1.
(86) Li, L.; Wang, B.; Meroueh, S. O. Support Vector Regression Scoring of Receptor–Ligand Complexes for Rank-Ordering and Virtual Screening of Chemical Libraries. *J. Chem. Inf. Model.* **2011**, *51* (9), 2132–2138. https://doi.org/10.1021/ci200078f.
(87) Wójcikowski, M.; Ballester, P. J.; Siedlecki, P. Performance of Machine-Learning Scoring Functions in Structure-Based Virtual Screening. *Sci. Rep.* **2017**, *7* (1), 46710. https://doi.org/10.1038/srep46710.
(88) Ragoza, M.; Hochuli, J.; Idrobo, E.; Sunseri, J.; Koes, D. R. Protein–Ligand Scoring with Convolutional Neural Networks. *J. Chem. Inf. Model.* **2017**, *57* (4), 942–957. https://doi.org/10.1021/acs.jcim.6b00740.
(89) Yasuo, N.; Sekijima, M. Improved Method of Structure-Based Virtual Screening via Interaction-Energy-Based Learning. *J. Chem. Inf. Model.* **2019**, *59* (3), 1050–1061. https://doi.org/10.1021/acs.jcim.8b00673.
(90) Ashtawy, H. M.; Mahapatra, N. R. Task-Specific Scoring Functions for Predicting Ligand Binding Poses and Affinity and for Screening Enrichment. *J. Chem. Inf. Model.* **2018**, *58* (1), 119–133. https://doi.org/10.1021/acs.jcim.7b00309.
(91) Chaput, L.; Martinez-Sanz, J.; Saettel, N.; Mouawad, L. Benchmark of Four Popular Virtual Screening Programs: Construction of the Active/Decoy Dataset Remains a Major Determinant of Measured Performance. *J. Cheminformatics* **2016**, *8* (1), 56. https://doi.org/10.1186/s13321-016-0167-x.
(92) Baum, B.; Muley, L.; Smolinski, M.; Heine, A.; Hangauer, D.; Klebe, G. Non-Additivity of Functional Group Contributions in Protein–Ligand Binding: A Comprehensive Study by Crystallography and Isothermal Titration Calorimetry. *J. Mol. Biol.* **2010**, *397* (4), 1042–1054. https://doi.org/10.1016/j.jmb.2010.02.007.
(93) Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45* (1), 5–32. https://doi.org/10.1023/A:1010933404324.
(94) Ballester, P. J.; Schreyer, A.; Blundell, T. L. Does a More Precise Chemical Description of Protein–Ligand Complexes Lead to More Accurate Prediction of Binding Affinity? *J. Chem. Inf. Model.* **2014**, *54* (3), 944–955. https://doi.org/10.1021/ci500091r.
(95) Wang, C.; Zhang, Y. Improving Scoring-Docking-Screening Powers of Protein-Ligand Scoring Functions Using Random Forest. *J. Comput. Chem.* **2017**, *38* (3), 169–177. https://doi.org/10.1002/jcc.24667.
(96) Ballester, P. J.; Mangold, M.; Howard, N. I.; Robinson, R. L. M.; Abell, C.; Blumberger, J.; Mitchell, J. B. O. Hierarchical Virtual Screening for the Discovery of New Molecular Scaffolds in Antibacterial Hit Identification. *J. R. Soc. Interface* **2012**, *9* (77), 3196–3207. https://doi.org/10.1098/rsif.2012.0569.
(97) Cortes, C.; Vapnik, V. Support-Vector Networks. *Mach. Learn.* **1995**, *20* (3), 273–297.

https://doi.org/10.1007/BF00994018.
(98) Das, S.; Krein, M. P.; Breneman, C. M. Binding Affinity Prediction with Property-Encoded Shape Distribution Signatures. *J. Chem. Inf. Model.* **2010**, *50* (2), 298–308. https://doi.org/10.1021/ci9004139.
(99) Ding, B.; Wang, J.; Li, N.; Wang, W. Characterization of Small Molecule Binding. I. Accurate Identification of Strong Inhibitors in Virtual Screening. *J. Chem. Inf. Model.* **2013**, *53* (1), 114–122. https://doi.org/10.1021/ci300508m.
(100) Xu, D.; Meroueh, S. O. Effect of Binding Pose and Modeled Structures on SVMGen and GlideScore Enrichment of Chemical Libraries. *J. Chem. Inf. Model.* **2016**, *56* (6), 1139–1151. https://doi.org/10.1021/acs.jcim.5b00709.
(101) Yan, Y.; Wang, W.; Sun, Z.; Zhang, J. Z. H.; Ji, C. Protein–Ligand Empirical Interaction Components for Virtual Screening. *J. Chem. Inf. Model.* **2017**, *57* (8), 1793–1806. https://doi.org/10.1021/acs.jcim.7b00017.
(102) Friedman, J. H. Greedy Function Approximation: A Gradient Boosting Machine. *Ann. Stat.* **2001**, *29* (5). https://doi.org/10.1214/aos/1013203451.
(103) Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. **2016**. https://doi.org/10.48550/ARXIV.1603.02754.
(104) Wang, B.; Zhao, Z.; Nguyen, D. D.; Wei, G.-W. Feature Functional Theory–Binding Predictor (FFT–BP) for the Blind Prediction of Binding Free Energies. *Theor. Chem. Acc.* **2017**, *136* (4), 55. https://doi.org/10.1007/s00214-017-2083-1.
(105) Lu, J.; Hou, X.; Wang, C.; Zhang, Y. Incorporating Explicit Water Molecules and Ligand Conformation Stability in Machine-Learning Scoring Functions. *J. Chem. Inf. Model.* **2019**, *59* (11), 4540–4549. https://doi.org/10.1021/acs.jcim.9b00645.
(106) Li, H.; Peng, J.; Sidorov, P.; Leung, Y.; Leung, K.-S.; Wong, M.-H.; Lu, G.; Ballester, P. J. Classical Scoring Functions for Docking Are Unable to Exploit Large Volumes of Structural and Interaction Data. *Bioinformatics* **2019**, *35* (20), 3989–3995. https://doi.org/10.1093/bioinformatics/btz183.
(107) Adeshina, Y. O.; Deeds, E. J.; Karanicolas, J. Machine Learning Classification Can Reduce False Positives in Structure-Based Virtual Screening. *Proc. Natl. Acad. Sci.* **2020**, *117* (31), 18477–18488. https://doi.org/10.1073/pnas.2000585117.
(108) Rezaei, M. A.; Li, Y.; Wu, D.; Li, X.; Li, C. Deep Learning in Drug Design: Protein-Ligand Binding Affinity Prediction. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2022**, *19* (1), 407–417. https://doi.org/10.1109/TCBB.2020.3046945.
(109) Wang, K.; Zhou, R.; Li, Y.; Li, M. DeepDTAF: A Deep Learning Method to Predict Protein–Ligand Binding Affinity. *Brief. Bioinform.* **2021**, *22* (5), bbab072. https://doi.org/10.1093/bib/bbab072.
(110) Zheng, L.; Fan, J.; Mu, Y. OnionNet: A Multiple-Layer Intermolecular-Contact-Based Convolutional Neural Network for Protein–Ligand Binding Affinity Prediction. *ACS Omega* **2019**, *4* (14), 15956–15965. https://doi.org/10.1021/acsomega.9b01997.
(111) Wallach, I.; Dzamba, M.; Heifets, A. AtomNet: A Deep Convolutional Neural Network for Bioactivity Prediction in Structure-Based Drug Discovery. arXiv 2015. https://doi.org/10.48550/ARXIV.1510.02855.
(112) Hsieh, C.-H.; Li, L.; Vanhauwaert, R.; Nguyen, K. T.; Davis, M. D.; Bu, G.; Wszolek, Z. K.; Wang, X. Miro1 Marks Parkinson's Disease Subset and Miro1 Reducer Rescues Neuron Loss in Parkinson's Models. *Cell Metab.* **2019**, *30* (6), 1131-1140.e7. https://doi.org/10.1016/j.cmet.2019.08.023.
(113) Pereira, J. C.; Caffarena, E. R.; Dos Santos, C. N. Boosting Docking-Based Virtual Screening with Deep Learning. *J. Chem. Inf. Model.* **2016**, *56* (12), 2495–2506. https://doi.org/10.1021/acs.jcim.6b00355.

(114) Imrie, F.; Bradley, A. R.; Van Der Schaar, M.; Deane, C. M. Protein Family-Specific Models Using Deep Neural Networks and Transfer Learning Improve Virtual Screening and Highlight the Need for More Data. *J. Chem. Inf. Model.* **2018**, *58* (11), 2319–2330. https://doi.org/10.1021/acs.jcim.8b00350.
(115) Jiménez, J.; Škalič, M.; Martínez-Rosell, G.; De Fabritiis, G. K_{DEEP} : Protein–Ligand Absolute Binding Affinity Prediction via 3D-Convolutional Neural Networks. *J. Chem. Inf. Model.* **2018**, *58* (2), 287–296. https://doi.org/10.1021/acs.jcim.7b00650.
(116) Stepniewska-Dziubinska, M. M.; Zielenkiewicz, P.; Siedlecki, P. Development and Evaluation of a Deep Learning Model for Protein–Ligand Binding Affinity Prediction. *Bioinformatics* **2018**, *34* (21), 3666–3674. https://doi.org/10.1093/bioinformatics/bty374.
(117) Öztürk, H.; Özgür, A.; Ozkirimli, E. DeepDTA: Deep Drug–Target Binding Affinity Prediction. *Bioinformatics* **2018**, *34* (17), i821–i829. https://doi.org/10.1093/bioinformatics/bty593.
(118) McNutt, A. T.; Francoeur, P.; Aggarwal, R.; Masuda, T.; Meli, R.; Ragoza, M.; Sunseri, J.; Koes, D. R. GNINA 1.0: Molecular Docking with Deep Learning. *J. Cheminformatics* **2021**, *13*, 43. https://doi.org/10.1186/s13321-021-00522-2.
(119) McNutt, A. T.; Li, Y.; Meli, R.; Aggarwal, R.; Koes, D. R. GNINA 1.3: The next Increment in Molecular Docking with Deep Learning. *J. Cheminformatics* **2025**, *17* (1), 28. https://doi.org/10.1186/s13321-025-00973-x.
(120) Sunseri, J.; Koes, D. R. Virtual Screening with Gnina 1.0. *Molecules* **2021**, *26* (23), 7369. https://doi.org/10.3390/molecules26237369.
(121) Satorras, V. G.; Hoogeboom, E.; Welling, M. E(n) Equivariant Graph Neural Networks. arXiv 2021. https://doi.org/10.48550/ARXIV.2102.09844.
(122) Kearnes, S.; McCloskey, K.; Berndl, M.; Pande, V.; Riley, P. Molecular Graph Convolutions: Moving beyond Fingerprints. *J. Comput. Aided Mol. Des.* **2016**, *30* (8), 595–608. https://doi.org/10.1007/s10822-016-9938-8.
(123) Karlov, D. S.; Sosnin, S.; Fedorov, M. V.; Popov, P. graphDelta: MPNN Scoring Function for the Affinity Prediction of Protein–Ligand Complexes. *ACS Omega* **2020**, *5* (10), 5150–5159. https://doi.org/10.1021/acsomega.9b04162.
(124) Feinberg, E. N.; Sur, D.; Wu, Z.; Husic, B. E.; Mai, H.; Li, Y.; Sun, S.; Yang, J.; Ramsundar, B.; Pande, V. S. PotentialNet for Molecular Property Prediction. *ACS Cent. Sci.* **2018**, *4* (11), 1520–1530. https://doi.org/10.1021/acscentsci.8b00507.
(125) Son, J.; Kim, D. Development of a Graph Convolutional Neural Network Model for Efficient Prediction of Protein-Ligand Binding Affinities. *PLOS ONE* **2021**, *16* (4), e0249404. https://doi.org/10.1371/journal.pone.0249404.
(126) Li, S.; Zhou, J.; Xu, T.; Huang, L.; Wang, F.; Xiong, H.; Huang, W.; Dou, D.; Xiong, H. Structure-Aware Interactive Graph Neural Networks for the Prediction of Protein-Ligand Binding Affinity. arXiv 2021. https://doi.org/10.48550/ARXIV.2107.10670.
(127) Moon, S.; Zhung, W.; Yang, S.; Lim, J.; Kim, W. Y. PIGNet: A Physics-Informed Deep Learning Model toward Generalized Drug–Target Interaction Predictions. *Chem. Sci.* **2022**, *13* (13), 3661–3673. https://doi.org/10.1039/D1SC06946B.
(128) Moon, S.; Hwang, S.-Y.; Lim, J.; Kim, W. Y. PIGNet2: A Versatile Deep Learning-Based Protein–Ligand Interaction Prediction Model for Binding Affinity Scoring and Virtual Screening. *Digit. Discov.* **2024**, *3* (2), 287–299. https://doi.org/10.1039/D3DD00149K.
(129) Shen, C.; Zhang, X.; Deng, Y.; Gao, J.; Wang, D.; Xu, L.; Pan, P.; Hou, T.; Kang, Y. Boosting Protein–Ligand Binding Pose Prediction and Virtual Screening Based on Residue–Atom Distance Likelihood Potential and Graph Transformer. *J. Med. Chem.*

2022, *65* (15), 10691–10706. https://doi.org/10.1021/acs.jmedchem.2c00991.
(130) Cao, D.; Chen, G.; Jiang, J.; Yu, J.; Zhang, R.; Chen, M.; Zhang, W.; Chen, L.; Zhong, F.; Zhang, Y.; Lu, C.; Li, X.; Luo, X.; Zhang, S.; Zheng, M. Generic Protein–Ligand Interaction Scoring by Integrating Physical Prior Knowledge and Data Augmentation Modelling. *Nat. Mach. Intell.* **2024**, *6* (6), 688–700. https://doi.org/10.1038/s42256-024-00849-z.
(131) Dong, T.; Yang, Z.; Zhou, J.; Chen, C. Y.-C. Equivariant Flexible Modeling of the Protein–Ligand Binding Pose with Geometric Deep Learning. *J. Chem. Theory Comput.* **2023**, *19* (22), 8446–8459. https://doi.org/10.1021/acs.jctc.3c00273.
(132) Zhang, X.; Zhang, O.; Shen, C.; Qu, W.; Chen, S.; Cao, H.; Kang, Y.; Wang, Z.; Wang, E.; Zhang, J.; Deng, Y.; Liu, F.; Wang, T.; Du, H.; Wang, L.; Pan, P.; Chen, G.; Hsieh, C.-Y.; Hou, T. Efficient and Accurate Large Library Ligand Docking with KarmaDock. *Nat. Comput. Sci.* **2023**, *3* (9), 789–804. https://doi.org/10.1038/s43588-023-00511-5.
(133) Armstrong, M. S.; Morris, G. M.; Finn, P. W.; Sharma, R.; Moretti, L.; Cooper, R. I.; Richards, W. G. ElectroShape: Fast Molecular Similarity Calculations Incorporating Shape, Chirality and Electrostatics. *J. Comput. Aided Mol. Des.* **2010**, *24* (9), 789–801. https://doi.org/10.1007/s10822-010-9374-0.
(134) Kufareva, I.; Ilatovskiy, A. V.; Abagyan, R. Pocketome: An Encyclopedia of Small-Molecule Binding Sites in 4D. *Nucleic Acids Res.* **2012**, *40* (D1), D535–D540. https://doi.org/10.1093/nar/gkr825.
(135) Wierbowski, S. D.; Wingert, B. M.; Zheng, J.; Camacho, C. J. Cross-docking Benchmark for Automated Pose and Ranking Prediction of Ligand Binding. *Protein Sci.* **2020**, *29* (1), 298–305. https://doi.org/10.1002/pro.3784.
(136) Li, Y.; Han, L.; Liu, Z.; Wang, R. Comparative Assessment of Scoring Functions on an Updated Benchmark: 2. Evaluation Methods and General Results. *J. Chem. Inf. Model.* **2014**, *54* (6), 1717–1736. https://doi.org/10.1021/ci500081m.
(137) Mysinger, M. M.; Carchia, M.; Irwin, John. J.; Shoichet, B. K. Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking. *J. Med. Chem.* **2012**, *55* (14), 6582–6594. https://doi.org/10.1021/jm300687e.
(138) Shen, C.; Hu, X.; Gao, J.; Zhang, X.; Zhong, H.; Wang, Z.; Xu, L.; Kang, Y.; Cao, D.; Hou, T. The Impact of Cross-Docked Poses on Performance of Machine Learning Classifier for Protein–Ligand Binding Pose Prediction. *J. Cheminformatics* **2021**, *13* (1), 81. https://doi.org/10.1186/s13321-021-00560-w.
(139) Su, M.; Yang, Q.; Du, Y.; Feng, G.; Liu, Z.; Li, Y.; Wang, R. Comparative Assessment of Scoring Functions: The CASF-2016 Update. *J. Chem. Inf. Model.* **2019**, *59* (2), 895–913. https://doi.org/10.1021/acs.jcim.8b00545.
(140) Cheng, T.; Li, X.; Li, Y.; Liu, Z.; Wang, R. Comparative Assessment of Scoring Functions on a Diverse Test Set. *J. Chem. Inf. Model.* **2009**, *49* (4), 1079–1093. https://doi.org/10.1021/ci9000053.
(141) Li, Y.; Liu, Z.; Li, J.; Han, L.; Liu, J.; Zhao, Z.; Wang, R. Comparative Assessment of Scoring Functions on an Updated Benchmark: 1. Compilation of the Test Set. *J. Chem. Inf. Model.* **2014**, *54* (6), 1700–1716. https://doi.org/10.1021/ci500080q.
(142) Dunbar, J. B.; Smith, R. D.; Yang, C.-Y.; Ung, P. M.-U.; Lexa, K. W.; Khazanov, N. A.; Stuckey, J. A.; Wang, S.; Carlson, H. A. CSAR Benchmark Exercise of 2010: Selection of the Protein–Ligand Complexes. *J. Chem. Inf. Model.* **2011**, *51* (9), 2036–2046. https://doi.org/10.1021/ci200082t.
(143) Ahmed, A.; Smith, R. D.; Clark, J. J.; Dunbar, J. B.; Carlson, H. A. Recent Improvements to Binding MOAD: A Resource for Protein–Ligand Binding Affinities and Structures. *Nucleic Acids Res.* **2015**, *43* (D1), D465–D469.

https://doi.org/10.1093/nar/gku1088.
(144) Wu, Z.; Ramsundar, B.; Feinberg, E. N.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Leswing, K.; Pande, V. MoleculeNet: A Benchmark for Molecular Machine Learning. *Chem. Sci.* **2018**, *9* (2), 513–530. https://doi.org/10.1039/C7SC02664A.
(145) Huang, K.; Fu, T.; Gao, W.; Zhao, Y.; Roohani, Y.; Leskovec, J.; Coley, C. W.; Xiao, C.; Sun, J.; Zitnik, M. Therapeutics Data Commons: Machine Learning Datasets and Tasks for Drug Discovery and Development. arXiv 2021. https://doi.org/10.48550/ARXIV.2102.09548.
(146) Liu, Y.; Dong, H.; Wang, X.; Moretti, R.; Wang, Y.; Su, Z.; Gu, J.; Bodenheimer, B.; Weaver, C. D.; Meiler, J.; Derr, T. WelQrate: Defining the Gold Standard in Small Molecule Drug Discovery Benchmarking. arXiv 2024. https://doi.org/10.48550/ARXIV.2411.09820.
(147) Wallach, I.; Heifets, A. Most Ligand-Based Classification Benchmarks Reward Memorization Rather than Generalization. *J. Chem. Inf. Model.* **2018**, *58* (5), 916–932. https://doi.org/10.1021/acs.jcim.7b00403.
(148) Chen, L.; Cruz, A.; Ramsey, S.; Dickson, C. J.; Duca, J. S.; Hornak, V.; Koes, D. R.; Kurtzman, T. Hidden Bias in the DUD-E Dataset Leads to Misleading Performance of Deep Learning in Structure-Based Virtual Screening. *PLOS ONE* **2019**, *14* (8), e0220113. https://doi.org/10.1371/journal.pone.0220113.
(149) Huang, N.; Shoichet, B. K.; Irwin, J. J. Benchmarking Sets for Molecular Docking. *J. Med. Chem.* **2006**, *49* (23), 6789–6801. https://doi.org/10.1021/jm0608356.
(150) Verdonk, M. L.; Berdini, V.; Hartshorn, M. J.; Mooij, W. T. M.; Murray, C. W.; Taylor, R. D.; Watson, P. Virtual Screening Using Protein−Ligand Docking: Avoiding Artificial Enrichment. *J. Chem. Inf. Comput. Sci.* **2004**, *44* (3), 793–806. https://doi.org/10.1021/ci034289q.
(151) Good, A. C.; Oprea, T. I. Optimization of CAMD Techniques 3. Virtual Screening Enrichment Studies: A Help or Hindrance in Tool Selection? *J. Comput. Aided Mol. Des.* **2008**, *22* (3–4), 169–178. https://doi.org/10.1007/s10822-007-9167-2.
(152) Rohrer, S. G.; Baumann, K. Maximum Unbiased Validation (MUV) Data Sets for Virtual Screening Based on PubChem Bioactivity Data. *J. Chem. Inf. Model.* **2009**, *49* (2), 169–184. https://doi.org/10.1021/ci8002649.
(153) Vogel, S. M.; Bauer, M. R.; Boeckler, F. M. DEKOIS: Demanding Evaluation Kits for Objective *in Silico* Screening — A Versatile Tool for Benchmarking Docking Programs and Scoring Functions. *J. Chem. Inf. Model.* **2011**, *51* (10), 2650–2665. https://doi.org/10.1021/ci2001549.
(154) Cummings, M. D.; DesJarlais, R. L.; Gibbs, A. C.; Mohan, V.; Jaeger, E. P. Comparison of Automated Docking Programs as Virtual Screening Tools. *J. Med. Chem.* **2005**, *48* (4), 962–976. https://doi.org/10.1021/jm049798d.
(155) Bauer, M. R.; Ibrahim, T. M.; Vogel, S. M.; Boeckler, F. M. Evaluation and Optimization of Virtual Screening Workflows with DEKOIS 2.0 – A Public Library of Challenging Docking Benchmark Sets. *J. Chem. Inf. Model.* **2013**, *53* (6), 1447–1462. https://doi.org/10.1021/ci400115b.
(156) Stein, R. M.; Yang, Y.; Balius, T. E.; O'Meara, M. J.; Lyu, J.; Young, J.; Tang, K.; Shoichet, B. K.; Irwin, J. J. Property-Unmatched Decoys in Docking Benchmarks. *J. Chem. Inf. Model.* **2021**, *61* (2), 699–714. https://doi.org/10.1021/acs.jcim.0c00598.
(157) Tran-Nguyen, V.-K.; Jacquemard, C.; Rognan, D. LIT-PCBA: An Unbiased Data Set for Machine Learning and Virtual Screening. *J. Chem. Inf. Model.* **2020**, *60* (9), 4263–4273. https://doi.org/10.1021/acs.jcim.0c00155.
(158) Sieg, J.; Flachsenberg, F.; Rarey, M. In Need of Bias Control: Evaluating Chemical

Data for Machine Learning in Structure-Based Virtual Screening. *J. Chem. Inf. Model.* **2019**, *59* (3), 947–961. https://doi.org/10.1021/acs.jcim.8b00712.
(159) Mastropietro, A.; Pasculli, G.; Bajorath, J. Learning Characteristics of Graph Neural Networks Predicting Protein–Ligand Affinities. *Nat. Mach. Intell.* **2023**, *5* (12), 1427–1436. https://doi.org/10.1038/s42256-023-00756-9.
(160) Volkov, M.; Turk, J.-A.; Drizard, N.; Martin, N.; Hoffmann, B.; Gaston-Mathé, Y.; Rognan, D. On the Frustration to Predict Binding Affinities from Protein–Ligand Structures with Deep Neural Networks. *J. Med. Chem.* **2022**, *65* (11), 7946–7958. https://doi.org/10.1021/acs.jmedchem.2c00487.
(161) Li, J.; Guan, X.; Zhang, O.; Sun, K.; Wang, Y.; Bagni, D.; Head-Gordon, T. Leak Proof PDBBind: A Reorganized Dataset of Protein-Ligand Complexes for More Generalizable Binding Affinity Prediction. *ArXiv* **2023**, arXiv:2308.09639v1.
(162) Francoeur, P. G.; Masuda, T.; Sunseri, J.; Jia, A.; Iovanisci, R. B.; Snyder, I.; Koes, D. R. Three-Dimensional Convolutional Neural Networks and a Cross-Docked Data Set for Structure-Based Drug Design. *J. Chem. Inf. Model.* **2020**, *60* (9), 4200–4215. https://doi.org/10.1021/acs.jcim.0c00411.
(163) Durant, G.; Boyles, F.; Birchall, K.; Marsden, B.; Deane, C. M. Robustly Interrogating Machine Learning-Based Scoring Functions: What Are They Learning? November 2, 2023. https://doi.org/10.1101/2023.10.30.564251.
(164) Zhu, H.; Yang, J.; Huang, N. Assessment of the Generalization Abilities of Machine-Learning Scoring Functions for Structure-Based Virtual Screening. *J. Chem. Inf. Model.* **2022**, *62* (22), 5485–5502. https://doi.org/10.1021/acs.jcim.2c01149.
(165) Boyles, F.; Deane, C. M.; Morris, G. M. Learning from Docked Ligands: Ligand-Based Features Rescue Structure-Based Scoring Functions When Trained on Docked Poses. *J. Chem. Inf. Model.* **2022**, *62* (22), 5329–5341. https://doi.org/10.1021/acs.jcim.1c00096.
(166) Scantlebury, J.; Vost, L.; Carbery, A.; Hadfield, T. E.; Turnbull, O. M.; Brown, N.; Chenthamarakshan, V.; Das, P.; Grosjean, H.; Von Delft, F.; Deane, C. M. A Small Step Toward Generalizability: Training a Machine Learning Scoring Function for Structure-Based Virtual Screening. *J. Chem. Inf. Model.* **2023**, *63* (10), 2960–2974. https://doi.org/10.1021/acs.jcim.3c00322.
(167) Wang, D.; Cui, C.; Ding, X.; Xiong, Z.; Zheng, M.; Luo, X.; Jiang, H.; Chen, K. Improving the Virtual Screening Ability of Target-Specific Scoring Functions Using Deep Learning Methods. *Front. Pharmacol.* **2019**, *10*, 924. https://doi.org/10.3389/fphar.2019.00924.

2 Development of Docked Pose Databases of PPI Inhibitor-Protein Complexes

2.1 Introduction

The development of data-driven methods in structure-based virtual screening (SBVS) relies heavily on available data, such as the crystal structure of the bound protein-ligand complex, and experimental binding constants. As discussed in Chapter 1, machine learning scoring functions (MLSFs) are trained on such data to predict three main tasks of molecular docking: binding affinity (scoring), binding pose (docking), or ranking (screening). Thus, these methods would benefit from large quantities of high-quality data available for training, as evidence supports that more data leads to improved predictive accuracies even without using more complex model architectures or engineered features[1,2]. Indeed, many MLSFs developed to date have taken advantage of the large and diverse sets of data from community benchmarks such as PDBbind[3] (further curated into CASF[4]), CSAR[5], BindingMOAD[6] (for structure and binding affinity data), DUD-E[7], LIT-PCBA[8], and DEKOIS2.0[9] (for structure and bioactivity data), to train and validate their models. In this thesis, rather than developing an MLSF for generic purposes, we aim to develop scoring functions (SFs) specifically for predicting interactions between protein-protein interaction (PPI) targets and their small-molecule inhibitors, which exhibit distinct physicochemical properties compared to conventional protein-ligand interactions[10].

We aim to propose MLSFs that specifically improve docking power, which refers to the ability of a SF to select the correct binding pose that closely imitates the native binding pose. This approach differs from the framework of the majority of MLSFs, which attempt to improve scoring or screening, and therefore indirectly improve docking and ranking, which are closely related tasks. The most common framework involves developing a model that,

given input data representing protein-ligand interactions, directly outputs a binding score for a given ligand and its pose. From there, docking power can be evaluated by examining the pose with the best docking score, ranking power by comparing the ranks determined by the docking scores, and screening power by comparing the docking scores of different ligands. We took a different approach and followed a procedure outlined by Ragoza et al.[11], which was adopted in several other recent studies[12–16]. Instead of being trained to predict binding affinity scores, these models are trained to output a probability between 0 and 1, indicating the likelihood of a pose resembling the native binding pose[16].

What are the reasons behind recasting essentially a regression problem (predicting binding scores) into a binary classification problem (predicting whether or not a given pose is correct/incorrect)? First and foremost, the accuracy of the regression models for the binding affinity prediction is inherently limited by the quality of the experimental data used for training. Some studies have found that commonly used bioactivity properties such as K_d, K_i, and IC_{50} are generally not reproducible and depend on various binding assay conditions even for the same complexes[17,18]. Therefore, training with datasets like PDBbind or BindingMOAD containing such noisy data would entail major limitations, as recently demonstrated by Landrum and Riniker[18]. Second, even if reliable binding affinity data exist, many of them are unavailable to the public as high-quality datasets are often proprietary and owned by biotechnology or pharmaceutical companies[19]. Third, an appropriate docking power in a SF is essential for screening, as some studies have hinted[11,14,20,21]. Virtual screening (VS), in which different ligands are ranked based on their predicted affinity to a given target, does not generally consider multiple poses from a single docking experiment. Therefore, if a SF cannot correctly identify the best pose from the ensemble of docked poses, comparing the docking scores of one ligand to others would hinder rather than pave the way

to the discovery of real actives. **Fig. 2.1** illustrates this critical need in SBVS endeavors.

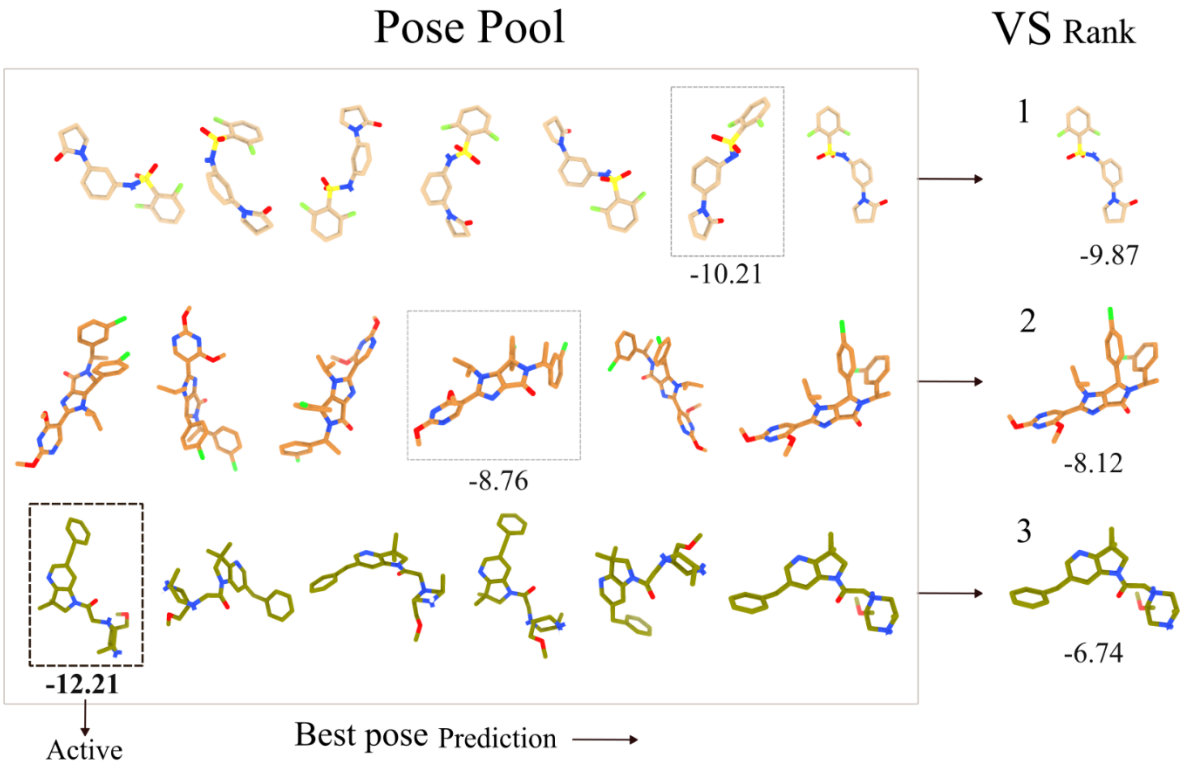


Figure 2.1: The impact of docking power on VS. Pose pool containing all ligand poses generated during docking. The rightmost pose in each row represents the pose prioritized by a given scoring method (lowest binding affinity), while the boxed poses represent the actual best pose in the ensemble. As illustrated, VS may fail to correctly rank and identify an active compound if the SF's pose prediction is suboptimal.

Given such considerations, developing MLSF models that can distinguish between correct and incorrect ligand poses offers an alternative strategy that avoids the experimental uncertainty of binding constants while simultaneously enhancing screening power indirectly.

2.2 Methods

To achieve the goal of developing MLSFs to improve docking power for PPI targets, the first phase of this project involved constructing a pose database of small-molecule PPI inhibitors docked into multiple binding pockets. The detailed procedures adapted to generate docked poses will be outlined in the following section.

2.2.1 2P2IDB: Structural Database of PPI Complexes and Their Inhibitors

2P2IDB is a hand-curated database containing 3D structures of protein-protein complexes with known orthosteric small-molecule inhibitors[22]. 2P2IDB was an excellent starting point towards the construction of our database since the entries are restricted to PPIs for which the structures of both the protein-protein and protein-inhibitor complexes were deposited[23,24] in RCSB Protein Data Bank (PDB)[25]. The entirety of the 2P2IDB dataset (version: 2023-09-14) was downloaded, containing 47 protein targets and 2,033 protein-ligand complexes (of which 1,717 are unique ligands). The database was filtered to remove any duplicates, entries with covalent bonds, ligands with molecular weight greater than 1,000 Daltons, and structures solved by nuclear magnetic resonance (NMR), resulting in 45 protein targets and 1,591 protein-ligand complexes (of which 1,576 are unique ligands). **Fig. 2.2** summarizes the relevant physicochemical properties of this filtered dataset.

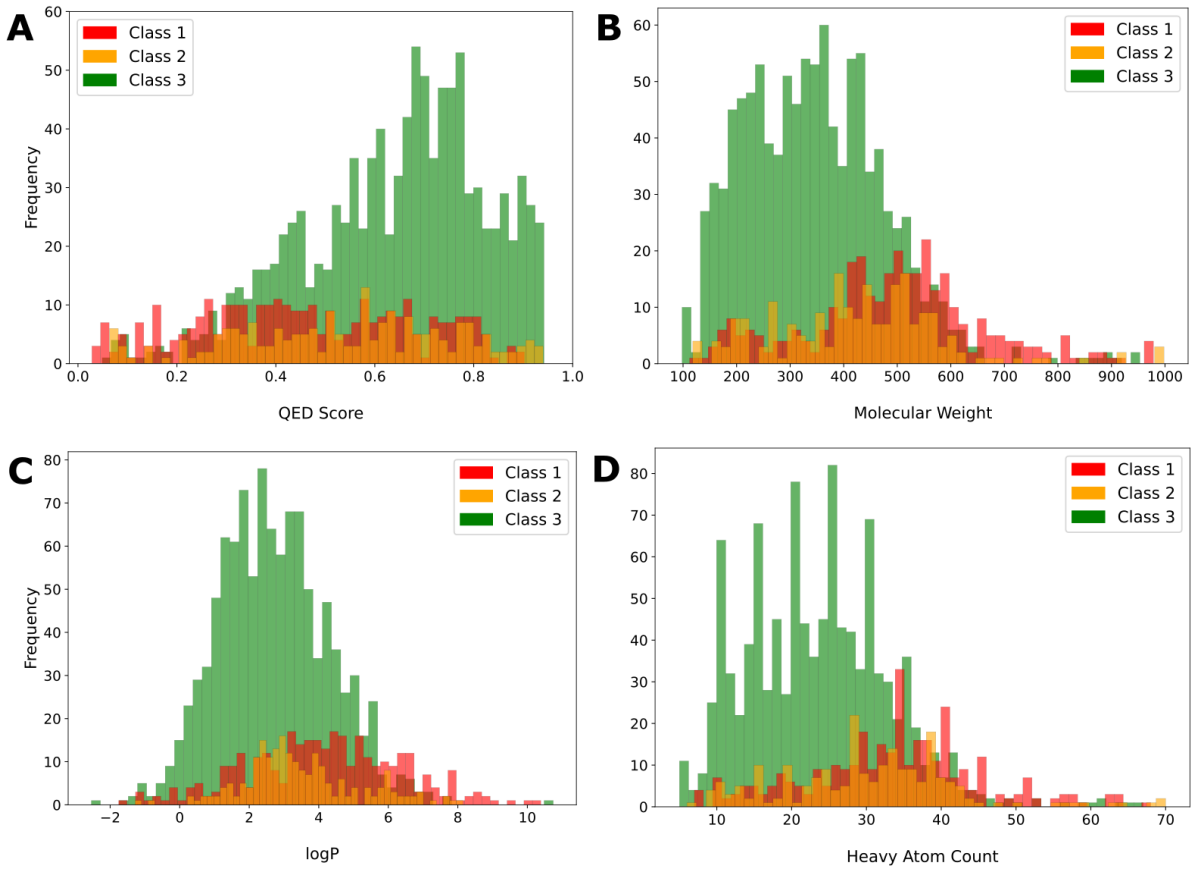


Figure 2.2: Distribution of relevant physiochemical properties of the inhibitors in the filtered 2P2IDB database. Class 1 consists of protein-peptide complexes, Class 2 consists of globular protein-protein complexes, and Class 3 consists of Bromodomain/Histone protein-protein complexes[22]. **A.** Quantitative estimate of drug-likeness (QED)[26] score. **B.** Molecular weight. **C.** Lipophilicity measured in LogP[27]. **D.** Heavy atom count.

2.2.2 Molecular Docking with AutoDock and GNINA

2.2.2.1 Choice of Docking Programs

To create a database consisting of docked poses for training and evaluation of docking power, we chose two independent docking programs based on different search and scoring algorithms. Our first choice was AutoDock-GPU (hereafter referred to as AutoDock), which is an OpenCL implementation of AutoDock4[28] to accelerate the docking runtime, chosen for its speed as well as wide usage across computational drug discovery[29]. It implements a Lamarckian genetic algorithm to search through the ligand conformation space for pose generation, and utilizes a force field-based SF to estimate binding affinities[28]. Our second choice was GNINA 1.0 (hereafter referred to as GNINA), which is a successor of SMINA[30]

and AutoDock Vina[31]. GNINA combines Monte Carlo sampling for pose generation and convolutional neural networks (CNNs) as a SF[32]. It was chosen because it is a recently developed MLSF designed as a multitask model that jointly predicts binding affinity ("CNNaffinity") and pose probability ("CNNscore"), the latter of which can be directly used to evaluate docking power[32]. Moreover, GNINA has recently emerged as the top-performing ML-based program in a series of unbiased drug discovery challenges[33].

Unlike GNINA, AutoDock simply outputs a docking score in kcal/mol, which was used to indirectly calculate metrics for docking power[28]. It is worthwhile to note that both programs are open-source projects, with relative ease of use.

2.2.2.2 Protein and Ligand Preparation

Pre-filtered 2P2IDB protein-ligand complexes were downloaded directly from the PDB database in PDB format. PDB files were first stripped of any solvent, ion, and water molecules, and the chain involved in protein-ligand interaction (as specified in the database) was extracted via pdb-tools[34]. In order to prepare the complex for molecular docking, Molecular Operating Environment (MOE)[35] software was used with KNIME[36] to automate the process for all 1591 complexes. All complexes were minimized, protonated at physiological pH (7) and corrected of any structural defects with MOE's QuickPrep module, and chains belonging to the same proteins were structurally superposed. Finally, the ligand files in PDB format were converted into SMILES strings and back into PDB with OpenBabel[37] software to randomize the initial ligand conformation and remove any potential bias of docking programs during the pose generation step.

2.2.2.3 Modes of Docking

We applied three modes of molecular docking to generate docked poses with AutoDock and GNINA. Re-docking refers to docking a ligand into a cognate binding

site/protein, by simply taking the protein and ligand from a known complex structure and docking the ligand to its pocket. However, binding poses generated via re-docking do not simulate the scoring and screening scenarios in real-world applications, in which ligands are docked into novel proteins with non-cognate pockets. A common practice to generate more realistic docking poses is by cross-docking, wherein ligands are docked into proteins that are similar to their cognate proteins[12]. For example, Francoeur et al. carried out cross-docking by utilizing Pocketome[38], which groups together protein structures with multiple similar binding sites, and docking ligands across the grouped pockets to create CrossDocked2020 database[12]. Shen et al. applied a similar method by clustering proteins in PDBbind dataset by their sequence similarity and cross-docking within the cluster to create PDBbind-CrossDocked-Core dataset[16]. Closely following their procedures, we performed cross-docking across a group of proteins (identified by the same UniProt ID, or PDB ID of the protein-protein complex) co-crystallized with different ligands in the filtered 2P2IDB dataset. Ligand-induced variations in protein conformation created non-cognate structures better suited for cross-docking. It is worthwhile to note that 5 of the 45 targets only had only one co-crystallized ligand, therefore it was not possible to create cross-docked poses for those targets (UniProt[40] ID: Q09472, Q8IZX4, Q9BXF3, Q9ULI0, Q9UPN9; Gene: EP300, TAF1L-2, CECR2, KIAA1240, TRIM33).

In addition to the above two modes of docking, we generated another dataset by docking ligands into the protein which was taken from the bound conformation of the protein-protein complex. The last dataset was added to benchmark the pose generation capability of AutoDock and GNINA in a more challenging scenario imitating VS against new PPI targets, i.e., docking against a PPI interface that has not been solved in complex with a small molecule. All three modes of docking are illustrated in **Fig. 2.3**.

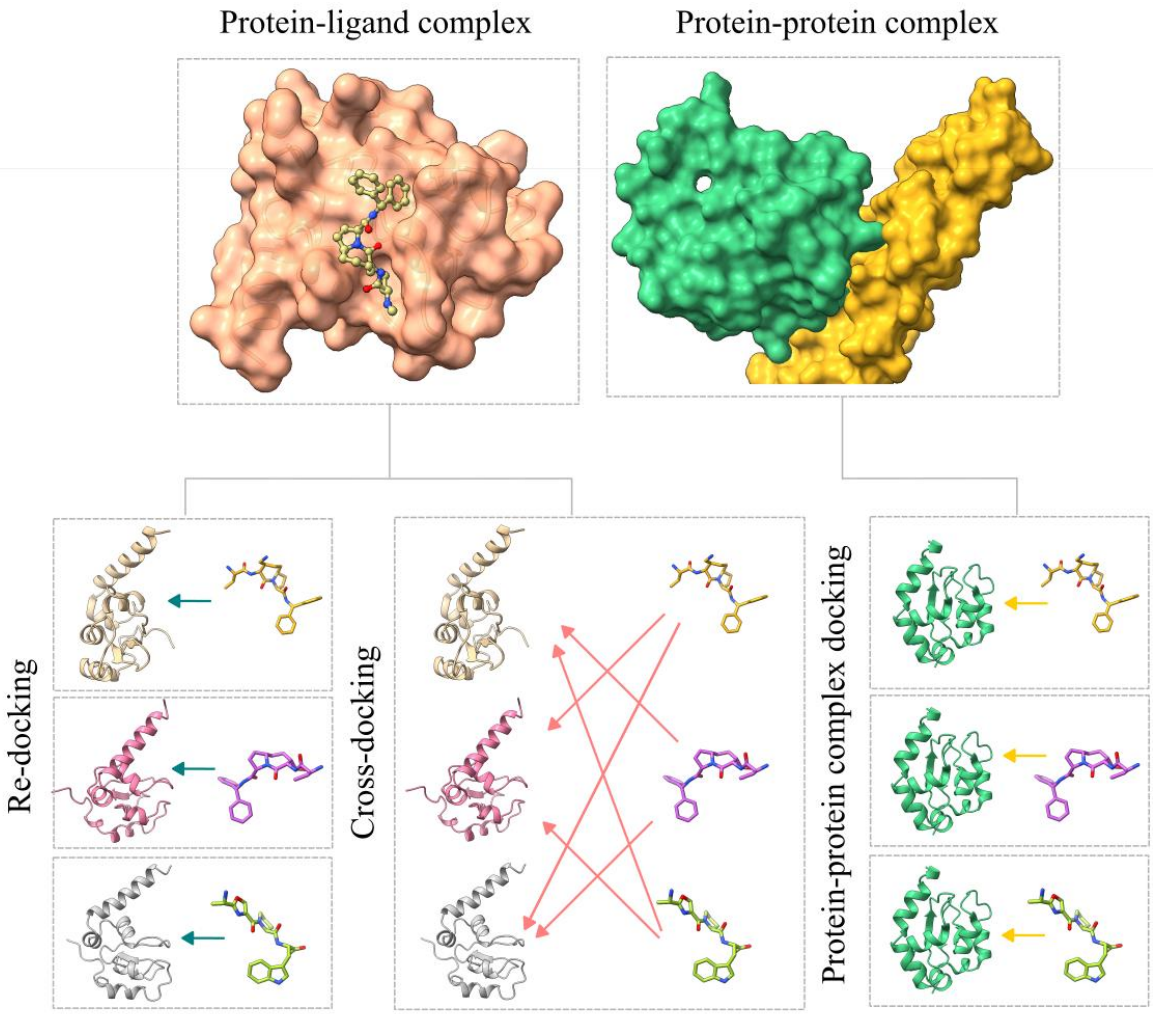


Figure 2.3: Three modes of docking. In re-docking, ligands are docked into the pocket of protein from the co-crystallized protein-ligand structure. In cross-docking, ligands are docked into the pocket of the same protein, which was not co-crystallized with them. In protein-protein pocket docking, ligands are docked into the pocket of the same protein, which was taken from the protein-protein complex.

For both AutoDock and GNINA, the docking grid was set by fixing a box of size 18.75 x 18.75 x 18.75 Å at the center of mass of each ligand, which was calculated with Biopython's Structure module[41]. For both programs, up to 50 poses were generated with all other parameters set to default. AutoDock by default groups generated poses into clusters with a tolerance of 2 Å, so we extracted only the top-scoring pose from each cluster to avoid including overly similar poses in our database. To maintain consistency with this approach, we adjusted the GNINA setting *--min_rmsd_filter 2* to ensure that all output poses differed by a tolerance of 2 Å. Because AutoDock clustering procedure outputs different numbers of

clusters depending on the system while the number of GNINA poses is fixed, our final database contained significantly fewer AutoDock poses compared to GNINA poses.

2.3 Results and Discussion

2.3.1 Docked Pose Databases of PPI Inhibitors

Based on the above procedures, we generated six databases, each defined by the combination of docking mode and docking program employed. In total, AutoDock ReDocked contained 1,329 complexes and 8,550 poses, AutoDock CrossDocked contained 181,498 complexes and 1,139,925 poses, AutoDock Protein-Protein PocketDocked contained 1,170 complexes and 11,411 poses, GNINA ReDocked contained 1,420 complexes and 61,281 poses, and GNINA CrossDocked contained 144,576 complexes and 6,226,487 poses, and GNINA Protein-Protein PocketDocked contained 1,180 complexes and 52,151 poses. Note that from the initial 1,591 protein-ligand complexes, some docking simulations failed for various reasons resulting in different numbers of complexes across the databases as a starting point for docking.

To classify a pose as structurally consistent with the native binding pose from incorrect poses, root mean square deviation (RMSD) values were calculated for each pose in the databases using a Python module spyrmsd[42], omitting all hydrogen atoms. A pose was classified as a positive sample if the calculated RMSD was below a predefined cutoff value, otherwise it was classified as a negative sample. In this study, we selected two cutoff values: 2 Å, as it is standard in most docking studies, and 3 Å, to allow for a larger tolerance and thus increase the number of positive samples. It is noteworthy that both values have been regularly used in similar studies[4,13]. **Table 2.1** summarizes the databases as well as positives and negatives classified using two cutoff values.

Table 2.1: Summary of PPI inhibitor pose databases.

Data Set	Complexes	Poses	RMSD: 3 Å		RMSD: 2 Å	
			Positives	Negatives	Positives	Negatives
AutoDock ReDocked	1329	8550	494	8056	217	8333
AutoDock CrossDocked	181498	1139925	39787	1100138	15980	1123945
AutoDock Protein-Protein PocketDocked	1170	11411	406	11005	126	11285
GNINA ReDocked	1420	61281	3504	57777	1061	60220
GNINA CrossDocked	144576	6226487	290364	5936123	82758	6143729
GNINA Protein-Protein PocketDocked	1180	52151	1474	50677	371	51780

As can be seen in **Fig. 2.4**, AutoDock generated a slightly higher proportion of positive poses in most of the data sets, regardless of the cutoff used. As expected, both docking programs were less effective in generating near-native poses in the protein-protein pocket-docked scenario compared to cross-docking scenario, and cross-docking compared to re-docking. Notably, increasing the cutoff value from 2 Å to 3 Å at least doubled or tripled the proportion of positives in the databases.

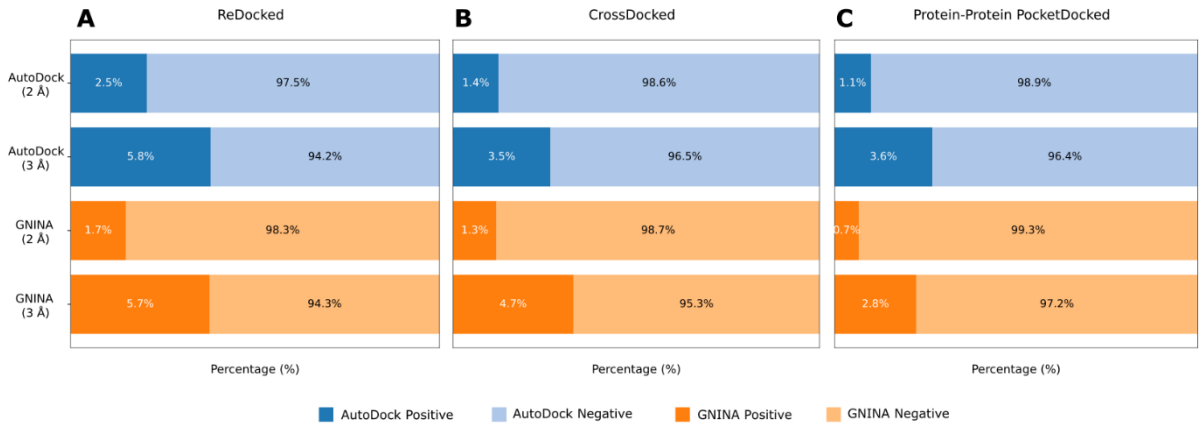


Figure 2.4: Imbalance in docked pose databases for PPI inhibitors**. A**. ReDocked sets. **B**. CrossDocked sets. **C**. Protein-Protein PocketDocked sets. Positive bars (in blue and orange) were scaled up for the sake of visualization.

2.3.2 Docking Power Benchmark

Next, the docking power of AutoDock and GNINA was benchmarked on PPI inhibitor pose databases. As mentioned in Section 1.3.2.1, a common metric for docking power is success rate in top-N poses (**Eq. 2.1**).

$$\mathrm{SR_{TopN}} = \frac{\#\ \text{of systems with a good pose ranked in top N}}{\text{total \# of docked systems}} \times 100\ (\%) \qquad (2.1)$$

In our study, we evaluated top-1 success rate (SR_1) as well as the overall success rate (N equal to all poses). SR_1 represents the proportion of the database that successfully generated a pose below a cutoff RMSD value, considering only the best pose as prioritized by a chosen docking or rescoring method. For AutoDock, this refers to the lowest binding energy pose, and for GNINA, it refers to the highest CNNscore pose. SR_1 will hereafter be referred to as Best Pose success rate. On the other hand, the overall success rate represents the proportion of the database where at least one of the docking-generated poses of the ensemble was below the selected cutoff RMSD value. Overall success rate will hereafter be referred to as All Poses success rate. While Best Pose success rate is related to the docking power, i.e. pose prediction ability of a SF, All Poses success rate is related to the pose generation, i.e. sampling power of a SF. **Fig. 2.5** illustrates how each system may be evaluated to calculate Best Pose success rate and All Poses success rate across the databases.

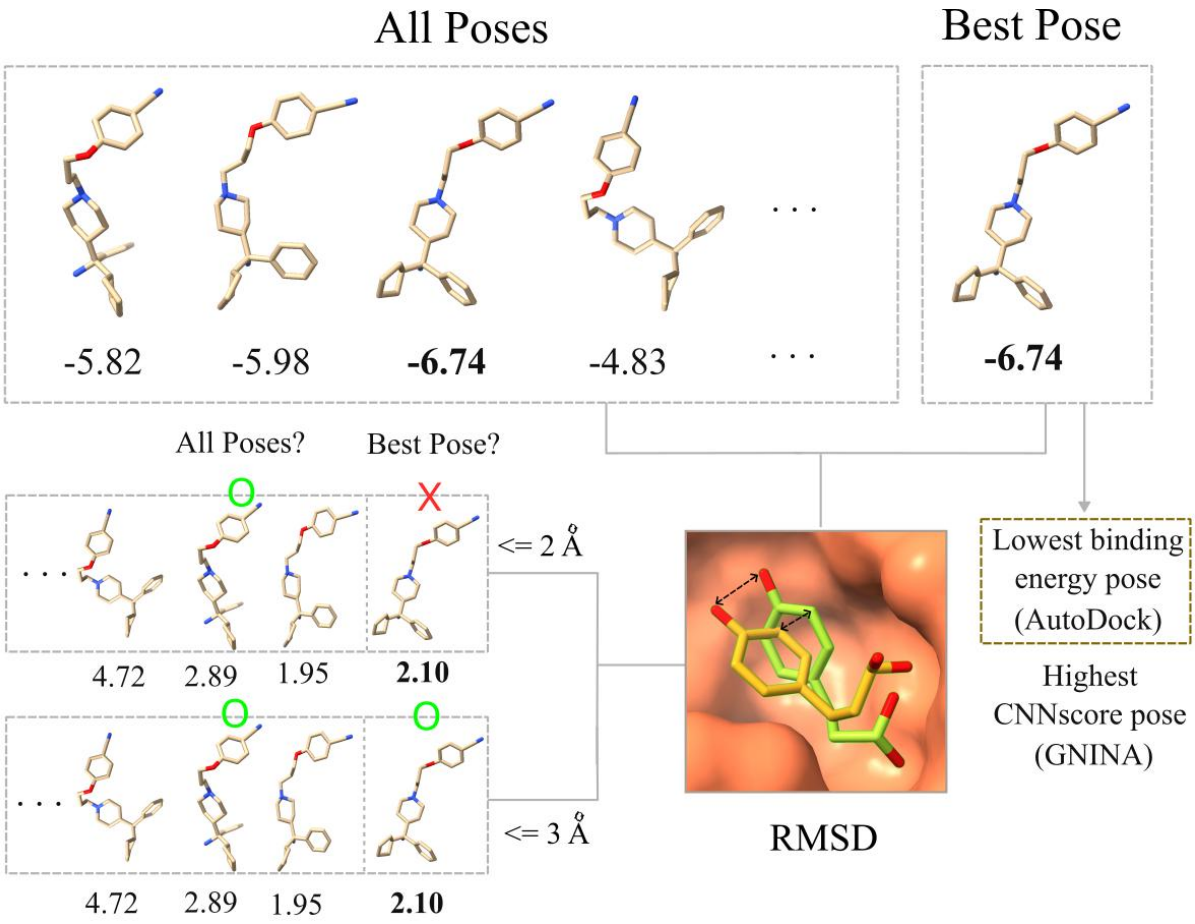


Figure 2.5: An example illustrating how a single system is evaluated for its docking success. On top, the boxes show all poses and best pose from the ensemble generated using AutoDock, with binding affinities shown below. RMSD values are computed for all poses scored during the docking, shown below each pose. Using 2 Å, the system failed to generate a good pose for Best Pose scenario but was successful in All Poses. Using 3 Å, the system is successful in both All Poses and Best Pose scenarios.

Success rates for both programs were calculated over all targets in the databases generated in this work. **Fig. 2.6** summarizes the docking power measured in Best Pose success rates and All Poses success rates for three cases: ReDocked, CrossDocked, and Protein-Protein Pocket-Docked sets as described previously.

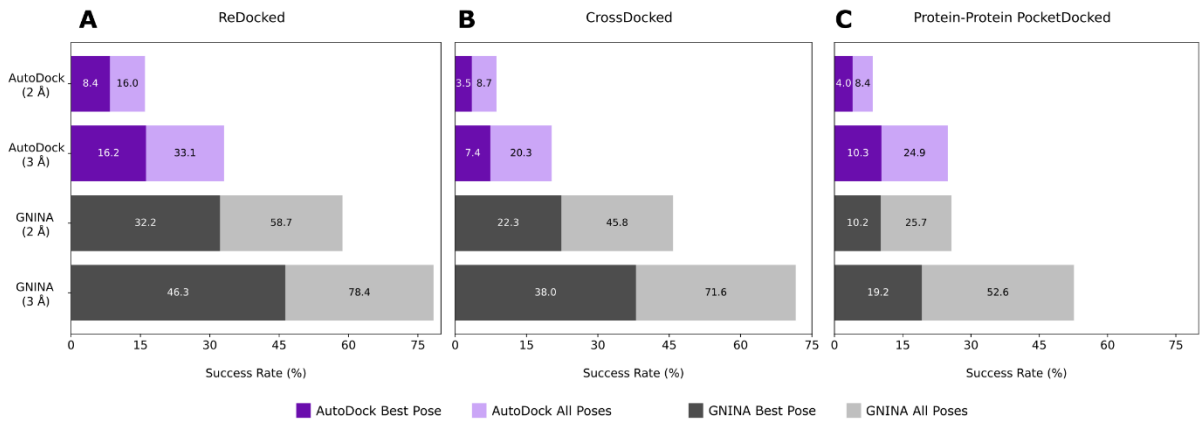


Figure 2.6: Best Pose and All Poses success rates across all targets in PPI inhibitor pose databases**. A**. ReDocked, **B**. CrossDocked, and **C**. Protein-Protein PocketDocked.

As shown in **Fig. 2.6-A-C**, GNINA consistently outperforms AutoDock by a large margin. The highest difference observed between GNINA and AutoDock was in the case of CrossDocked set with a cutoff value of 3 Å, where Best Pose success rate of GNINA was roughly five times higher than that of AutoDock (38.0% vs. 7.4%). GNINA's superior performances can be attributed to (1) an improved algorithm for pose generation, as GNINA uses a Monte Carlo sampling algorithm adopted by later generation of classical SFs (such as Autodock Vina[31]) and (2) GNINA's CNN-based MLSF directly trained to predict pose probability. Notably, while AutoDock performances were significantly lower across all datasets, the difference on the CrossDocked and Protein-Protein PocketDocked dataset was less significant. As success rates differed widely across different targets in our databases, per-target success rates were also computed and included in the appendix of this chapter (**Fig. 2.7**).

Next, we compared GNINA's docking power on the PPI inhibitor dataset to the original, larger and more general CrossDocked2020 dataset. The authors of GNINA used a cross-docking database by Wierbowski et al., which is composed of a diverse subset of targets from DUD-E database and applied similar strategy to create cross-docking set as that of PDBbind-CrossDocked-Core (described in Section 2.2.2.3)[43]. They reported Best Pose

success rate (RMSD <= 2 Å) of up to 42%, and All Poses success rate of 52-53% using the default ensemble CNN model, which was used in our docking experiments[32]. Comparing these results to our PPI CrossDocked set with the same cutoff value of 2 Å (Best Pose success rate of 22.3%, All Poses success rate of 45.8%), we observed a significant decrease in performances on PPI targets in cross-docked scenarios, and even worse on the more challenging Protein-Protein PocketDocked set (Best Pose success rate of 10.2% and All Poses success rate of 25.7%).

The purpose of computing All Poses success rate and comparing it to Best Pose success rate was to disconnect pose generation capability of docking programs from their docking power (**Fig. 2.6**). All Poses success rate marks the upper limit of a new SF, since it is obviously impossible to rescore poses that were not generated, which will be the case of roughly 45.8 % of systems in GNINA CrossDocked set with RMSD <= 2 Å, or 71.6 % with RMSD <= 3 Å. It is also clear that it is more meaningful to compare Best Pose success rates of different SFs starting from the same pool of generated poses. More importantly, in all datasets presented here, Best Pose success rates were lower than All Poses success rates, indicating a significant gap between sampling and docking power that can be potentially bridged using methods improving upon current SFs. Lastly, it is worthwhile to note that using a cutoff of 3 Å with GNINA resulted in good All Poses success rate, with an increase of almost 25% (45.8% to 71.6%) compared to using a cutoff of 2 Å. Based on the results of benchmarking, we decided to use labels generated from both cutoff values to train MLSFs, since 2 Å is a more broadly accepted value, but 3 Å enabled us to reduce the imbalance in the datasets while still being structurally relevant.

2.4 Conclusions

A new set of databases consisting of docked poses of PPI inhibitors was generated for

the purpose of training and validating MLSFs with enhanced docking power against this relevant but challenging class of therapeutic targets. A major contribution of this work is the development of the GNINA CrossDocked and Protein-Protein PocketDocked sets. As the first resources specifically focused on PPIs, these databases will be valuable for benchmarking established and emerging computational methods for tasks related to PPI modulator discovery. These databases are also suitable for developing MLSFs as they provide a large dataset of positive and negative samples obtained with cross-docking, thus resembling realistic docking and screening scenarios. Thus, the compiled datasets represent a significant step ahead in the development of new methodologies for accelerating PPI-based drug discovery.

Benchmarking two well-established docking methods on the developed databases clearly demonstrated that the docking power of current SFs, whether ML-based or not, can be further improved, as there is a significant gap between the capacity of docking programs to generate and prioritize near-native poses in PPI complexes. For example, GNINA CrossDocked set has an improvement potential reaching up to 45% for 2 Å and 71% for 3 Å, while its current MLSFs only performs at 22% and 38%, respectively. These results motivated us to develop improved MLSFs for enhanced docking power on PPI targets, as discussed in Chapter 3.

2.5 References

(1) Li, H.; Peng, J.; Sidorov, P.; Leung, Y.; Leung, K.-S.; Wong, M.-H.; Lu, G.; Ballester, P. J. Classical Scoring Functions for Docking Are Unable to Exploit Large Volumes of Structural and Interaction Data. *Bioinformatics* **2019**, *35* (20), 3989–3995. https://doi.org/10.1093/bioinformatics/btz183.

(2) Wójcikowski, M.; Ballester, P. J.; Siedlecki, P. Performance of Machine-Learning Scoring Functions in Structure-Based Virtual Screening. *Sci. Rep.* **2017**, *7* (1), 46710. https://doi.org/10.1038/success rateep46710.

(3) Wang, R.; Fang, X.; Lu, Y.; Wang, S. The PDBbind Database: Collection of Binding Affinities for Protein−Ligand Complexes with Known Three-Dimensional Structures. *J. Med. Chem.* **2004**, *47* (12), 2977–2980. https://doi.org/10.1021/jm030580l.

(4) Su, M.; Yang, Q.; Du, Y.; Feng, G.; Liu, Z.; Li, Y.; Wang, R. Comparative Assessment of Scoring Functions: The CASF-2016 Update. *J. Chem. Inf. Model.* **2019**, *59* (2), 895–913. https://doi.org/10.1021/acs.jcim.8b00545.

(5) Dunbar, J. B.; Smith, R. D.; Yang, C.-Y.; Ung, P. M.-U.; Lexa, K. W.; Khazanov, N. A.; Stuckey, J. A.; Wang, S.; Carlson, H. A. CSAR Benchmark Exercise of 2010: Selection of the Protein–Ligand Complexes. *J. Chem. Inf. Model.* **2011**, *51* (9), 2036–2046. https://doi.org/10.1021/ci200082t.

(6) Ahmed, A.; Smith, R. D.; Clark, J. J.; Dunbar, J. B.; Carlson, H. A. Recent Improvements to Binding MOAD: A Resource for Protein–Ligand Binding Affinities and Structures. *Nucleic Acids Res.* **2015**, *43* (D1), D465–D469. https://doi.org/10.1093/nar/gku1088.

(7) Mysinger, M. M.; Carchia, M.; Irwin, John. J.; Shoichet, B. K. Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking. *J. Med. Chem.* **2012**, *55* (14), 6582–6594. https://doi.org/10.1021/jm300687e.

(8) Tran-Nguyen, V.-K.; Jacquemard, C.; Rognan, D. LIT-PCBA: An Unbiased Data Set for Machine Learning and Virtual Screening. *J. Chem. Inf. Model.* **2020**, *60* (9), 4263–4273. https://doi.org/10.1021/acs.jcim.0c00155.

(9) Bauer, M. R.; Ibrahim, T. M.; Vogel, S. M.; Boeckler, F. M. Evaluation and Optimization of Virtual Screening Workflows with DEKOIS 2.0 – A Public Library of Challenging Docking Benchmark Sets. *J. Chem. Inf. Model.* **2013**, *53* (6), 1447–1462. https://doi.org/10.1021/ci400115b.

(10) Smith, M. C.; Gestwicki, J. E. Features of Protein–Protein Interactions That Translate into Potent Inhibitors: Topology, Surface Area and Affinity. *Expert Rev. Mol. Med.* **2012**, *14*, e16. https://doi.org/10.1017/erm.2012.10.

(11) Ragoza, M.; Hochuli, J.; Idrobo, E.; Sunseri, J.; Koes, D. R. Protein–Ligand Scoring with Convolutional Neural Networks. *J. Chem. Inf. Model.* **2017**, *57* (4), 942–957. https://doi.org/10.1021/acs.jcim.6b00740.

(12) Francoeur, P. G.; Masuda, T.; Sunseri, J.; Jia, A.; Iovanisci, R. B.; Snyder, I.; Koes, D. R. Three-Dimensional Convolutional Neural Networks and a Cross-Docked Data Set for Structure-Based Drug Design. *J. Chem. Inf. Model.* **2020**, *60* (9), 4200–4215. https://doi.org/10.1021/acs.jcim.0c00411.

(13) Jiang, H.; Fan, M.; Wang, J.; Sarma, A.; Mohanty, S.; Dokholyan, N. V.; Mahdavi, M.; Kandemir, M. T. Guiding Conventional Protein–Ligand Docking Software with Convolutional Neural Networks. *J. Chem. Inf. Model.* **2020**, *60* (10), 4594–4602. https://doi.org/10.1021/acs.jcim.0c00542.

(14) Morrone, J. A.; Weber, J. K.; Huynh, T.; Luo, H.; Cornell, W. D. Combining Docking

Pose Rank and Structure with Deep Learning Improves Protein–Ligand Binding Mode Prediction over a Baseline Docking Approach. *J. Chem. Inf. Model.* **2020**, *60* (9), 4170–4179. https://doi.org/10.1021/acs.jcim.9b00927.

(15) Scantlebury, J.; Vost, L.; Carbery, A.; Hadfield, T. E.; Turnbull, O. M.; Brown, N.; Chenthamarakshan, V.; Das, P.; Grosjean, H.; Von Delft, F.; Deane, C. M. A Small Step Toward Generalizability: Training a Machine Learning Scoring Function for Structure-Based Virtual Screening. *J. Chem. Inf. Model.* **2023**, *63* (10), 2960–2974. https://doi.org/10.1021/acs.jcim.3c00322.

(16) Shen, C.; Hu, X.; Gao, J.; Zhang, X.; Zhong, H.; Wang, Z.; Xu, L.; Kang, Y.; Cao, D.; Hou, T. The Impact of Cross-Docked Poses on Performance of Machine Learning Classifier for Protein–Ligand Binding Pose Prediction. *J. Cheminformatics* **2021**, *13* (1), 81. https://doi.org/10.1186/s13321-021-00560-w.

(17) Kramer, C.; Kalliokoski, T.; Gedeck, P.; Vulpetti, A. The Experimental Uncertainty of Heterogeneous Public K_i Data. *J. Med. Chem.* **2012**, *55* (11), 5165–5173. https://doi.org/10.1021/jm300131x.

(18) Landrum, G. A.; Riniker, S. Combining IC_{50} or K_i Values from Different Sources Is a Source of Significant Noise. *J. Chem. Inf. Model.* **2024**, *64* (5), 1560–1567. https://doi.org/10.1021/acs.jcim.4c00049.

(19) Kramer, C.; Chodera, J.; Damm-Ganamet, K. L.; Gilson, M. K.; Günther, J.; Lessel, U.; Lewis, R. A.; Mobley, D.; Nittinger, E.; Pecina, A.; Schapira, M.; Walters, W. P. The Need for Continuing Blinded Pose- and Activity Prediction Benchmarks. *J. Chem. Inf. Model.* **2025**, *65* (5), 2180–2190. https://doi.org/10.1021/acs.jcim.4c02296.

(20) Xu, D.; Meroueh, S. O. Effect of Binding Pose and Modeled Structures on SVMGen and GlideScore Enrichment of Chemical Libraries. *J. Chem. Inf. Model.* **2016**, *56* (6), 1139–1151. https://doi.org/10.1021/acs.jcim.5b00709.

(21) Imrie, F.; Bradley, A. R.; Van Der Schaar, M.; Deane, C. M. Protein Family-Specific Models Using Deep Neural Networks and Transfer Learning Improve Virtual Screening and Highlight the Need for More Data. *J. Chem. Inf. Model.* **2018**, *58* (11), 2319–2330. https://doi.org/10.1021/acs.jcim.8b00350.

(22) Basse, M.-J.; Betzi, S.; Morelli, X.; Roche, P. 2P2IDB: The Protein Protein Interaction Inhibitor Database, 2016. https://2p2idb.marseille.inserm.fr/index.html#references.

(23) Basse, M. J.; Betzi, S.; Bourgeas, R.; Bouzidi, S.; Chetrit, B.; Hamon, V.; Morelli, X.; Roche, P. 2P2Idb: A Structural Database Dedicated to Orthosteric Modulation of Protein–Protein Interactions. *Nucleic Acids Res.* **2012**, *41* (D1), D824–D827. https://doi.org/10.1093/nar/gks1002.

(24) Basse, M.-J.; Betzi, S.; Morelli, X.; Roche, P. 2P2Idb v2: Update of a Structural Database Dedicated to Orthosteric Modulation of Protein–Protein Interactions. *Database* **2016**, *2016*, baw007. https://doi.org/10.1093/database/baw007.

(25) Berman, H. M. The Protein Data Bank. *Nucleic Acids Res.* **2000**, *28* (1), 235–242. https://doi.org/10.1093/nar/28.1.235.

(26) Bickerton, G. R.; Paolini, G. V.; Besnard, J.; Muresan, S.; Hopkins, A. L. Quantifying the Chemical Beauty of Drugs. *Nat. Chem.* **2012**, *4* (2), 90–98. https://doi.org/10.1038/nchem.1243.

(27) Hansch, C.; Björkroth, J. P.; Leo, A. Hydrophobicity and Central Nervous System Agents: On the Principle of Minimal Hydrophobicity in Drug Design. *J. Pharm. Sci.* **1987**, *76* (9), 663–687. https://doi.org/10.1002/jps.2600760902.

(28) Morris, G. M.; Huey, R.; Lindstrom, W.; Sanner, M. F.; Belew, R. K.; Goodsell, D. S.; Olson, A. J. AutoDock4 and AutoDockTools4: Automated Docking with Selective Receptor Flexibility. *J. Comput. Chem.* **2009**, *30* (16), 2785–2791.

https://doi.org/10.1002/jcc.21256.
(29) Santos-Martins, D.; Solis-Vasquez, L.; Tillack, A. F.; Sanner, M. F.; Koch, A.; Forli, S. Accelerating A UTO D OCK 4 with GPUs and Gradient-Based Local Search. *J. Chem. Theory Comput.* **2021**, *17* (2), 1060–1073. https://doi.org/10.1021/acs.jctc.0c01006.
(30) Koes, D. R.; Baumgartner, M. P.; Camacho, C. J. Lessons Learned in Empirical Scoring with Smina from the CSAR 2011 Benchmarking Exercise. *J. Chem. Inf. Model.* **2013**, *53* (8), 1893–1904. https://doi.org/10.1021/ci300604z.
(31) Trott, O.; Olson, A. J. AutoDock Vina: Improving the Speed and Accuracy of Docking with a New Scoring Function, Efficient Optimization, and Multithreading. *J. Comput. Chem.* **2010**, *31* (2), 455–461. https://doi.org/10.1002/jcc.21334.
(32) McNutt, A. T.; Francoeur, P.; Aggarwal, R.; Masuda, T.; Meli, R.; Ragoza, M.; Sunseri, J.; Koes, D. R. GNINA 1.0: Molecular Docking with Deep Learning. *J. Cheminformatics* **2021**, *13*, 43. https://doi.org/10.1186/s13321-021-00522-2.
(33) Dunn, I.; Pirhadi, S.; Wang, Y.; Ravindran, S.; Concepcion, C.; Koes, D. R. CACHE Challenge #1: Docking with GNINA Is All You Need. *J. Chem. Inf. Model.* **2024**, *64* (24), 9388–9396. https://doi.org/10.1021/acs.jcim.4c01429.
(34) Rodrigues, J. P. G. L. M.; Teixeira, J. M. C.; Trellet, M.; Bonvin, A. M. J. J. Pdb-Tools: A Swiss Army Knife for Molecular Structures. *F1000Research* **2018**, *7*, 1961. https://doi.org/10.12688/f1000research.17456.1.
(35) Molecular Operating Environment (MOE), 2025.
(36) Berthold, M. R.; Cebron, N.; Dill, F.; Gabriel, T. R.; Kötter, T.; Meinl, T.; Ohl, P.; Thiel, K.; Wiswedel, B. KNIME - the Konstanz Information Miner: Version 2.0 and Beyond. *ACM SIGKDD Explor. Newsl.* **2009**, *11* (1), 26–31. https://doi.org/10.1145/1656274.1656280.
(37) O'Boyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T.; Hutchison, G. R. Open Babel: An Open Chemical Toolbox. *J. Cheminformatics* **2011**, *3* (1), 33. https://doi.org/10.1186/1758-2946-3-33.
(38) Kufareva, I.; Ilatovskiy, A. V.; Abagyan, R. Pocketome: An Encyclopedia of Small-Molecule Binding Sites in 4D. *Nucleic Acids Res.* **2012**, *40* (D1), D535–D540. https://doi.org/10.1093/nar/gkr825.
(39) Schaller, D. A.; Christ, C. D.; Chodera, J. D.; Volkamer, A. Benchmarking Cross-Docking Strategies in Kinase Drug Discovery. *J. Chem. Inf. Model.* **2024**, *64* (23), 8848–8858. https://doi.org/10.1021/acs.jcim.4c00905.
(40) The UniProt Consortium; Bateman, A.; Martin, M.-J.; Orchard, S.; Magrane, M.; Adesina, A.; Ahmad, S.; Bowler-Barnett, E. H.; Bye-A-Jee, H.; Carpentier, D.; Denny, P.; Fan, J.; Garmiri, P.; Gonzales, L. J. D. C.; Hussein, A.; Ignatchenko, A.; Insana, G.; Ishtiaq, R.; Joshi, V.; Jyothi, D.; Kandasaamy, S.; Lock, A.; Luciani, A.; Luo, J.; Lussi, Y.; Marin, J. S. M.; Raposo, P.; Rice, D. L.; Santos, R.; Speretta, E.; Stephenson, J.; Totoo, P.; Tyagi, N.; Urakova, N.; Vasudev, P.; Warner, K.; Wijerathne, S.; Yu, C. W.-H.; Zaru, R.; Bridge, A. J.; Aimo, L.; Argoud-Puy, G.; Auchincloss, A. H.; Axelsen, K. B.; Bansal, P.; Baratin, D.; Batista Neto, T. M.; Blatter, M.-C.; Bolleman, J. T.; Boutet, E.; Breuza, L.; Gil, B. C.; Casals-Casas, C.; Echioukh, K. C.; Coudert, E.; Cuche, B.; De Castro, E.; Estreicher, A.; Famiglietti, M. L.; Feuermann, M.; Gasteiger, E.; Gaudet, P.; Gehant, S.; Gerritsen, V.; Gos, A.; Gruaz, N.; Hulo, C.; Hyka-Nouspikel, N.; Jungo, F.; Kerhornou, A.; Mercier, P. L.; Lieberherr, D.; Masson, P.; Morgat, A.; Paesano, S.; Pedruzzi, I.; Pilbout, S.; Pourcel, L.; Poux, S.; Pozzato, M.; Pruess, M.; Redaschi, N.; Rivoire, C.; Sigrist, C. J. A.; Sonesson, K.; Sundaram, S.; Sveshnikova, A.; Wu, C. H.; Arighi, C. N.; Chen, C.; Chen, Y.; Huang, H.; Laiho, K.; Lehvaslaiho, M.; McGarvey, P.; Natale, D. A.; Ross, K.; Vinayaka, C. R.; Wang, Y.; Zhang, J.

UniProt: The Universal Protein Knowledgebase in 2025. *Nucleic Acids Res.* **2025**, *53* (D1), D609–D617. https://doi.org/10.1093/nar/gkae1010.
(41) Cock, P. J. A.; Antao, T.; Chang, J. T.; Chapman, B. A.; Cox, C. J.; Dalke, A.; Friedberg, I.; Hamelryck, T.; Kauff, F.; Wilczynski, B.; De Hoon, M. J. L. Biopython: Freely Available Python Tools for Computational Molecular Biology and Bioinformatics. *Bioinformatics* **2009**, *25* (11), 1422–1423. https://doi.org/10.1093/bioinformatics/btp163.
(42) Meli, R.; Biggin, P. C. Spyrmsd: Symmetry-Corrected RMSD Calculations in Python. *J. Cheminformatics* **2020**, *12* (1), 49. https://doi.org/10.1186/s13321-020-00455-2.
(43) Wierbowski, S. D.; Wingert, B. M.; Zheng, J.; Camacho, C. J. Cross-docking Benchmark for Automated Pose and Ranking Prediction of Ligand Binding. *Protein Sci.* **2020**, *29* (1), 298–305. https://doi.org/10.1002/pro.3784.

2.6 Appendix

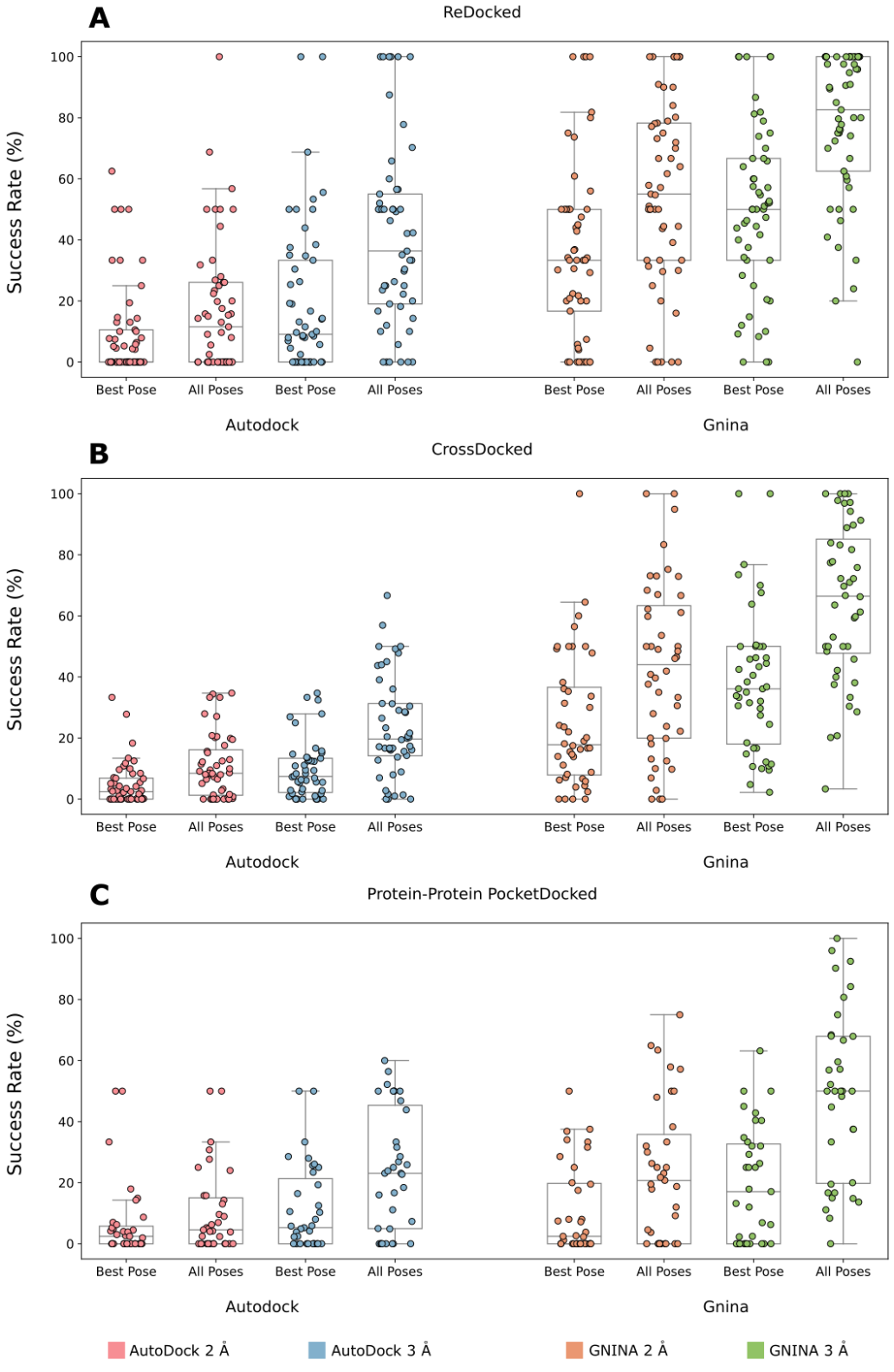


Figure 2.7: Best Pose and All Poses success rates of each target protein in PPI inhibitor pose databases. **A**. ReDocked. **B**. CrossDocked. **C**. Protein-Protein PocketDocked.

3 MLSFs for Prediction of Accurate Binding Poses for PPI Inhibitors

3.1 Introduction

In this chapter, we embark on achieving the second goal of this thesis, namely the development of machine learning scoring functions (MLSFs) to improve binding pose prediction of protein-protein interaction (PPI) inhibitors. Perhaps the most important considerations for designing machine learning (ML) models towards this goal are avoiding common pitfalls that lead to failure to generalize to systems unseen during training. As discussed in Section 1.3.3, two of the major pitfalls are (1) models that learn ligand and/or protein structures, by memorizing low-dimensional features (such as size of the ligand) rather than learning the underlying patterns in the intermolecular interaction between protein and ligand molecules and (2) data leakage leading to overoptimistic results on flawed retrospective benchmarks. We implement design choices that attempt to minimize the biases and data leakage to develop MLSFs that can be useful in prospective screening of novel PPI ligands.

We used the GNINA CrossDocked set described in Chapter 2 as the basis to build training and evaluation datasets. Cross-docked poses better imitate the intended application of our proposed MLSFs: predicting pose probability of ligands docked into an unknown pocket structure. Docking power benchmark from the previous chapter showed that GNINA's convolutional neural network (CNN)-based scoring function underperforms when applied to PPI databases. Furthermore, it was shown that GNINA's docking power can be improved within the constraints of its pose generation capabilities. While AutoDock faces the same issue, its pose generation performances were significantly lower in the PPI database to be useful in potential structure-based virtual screening (SBVS) applications, even if better

MLSFs were to be applied.

3.2 Methods

3.2.1 Preparation of Training and Evaluation Datasets

Recent studies have shown that commonly used data splitting strategies are inadequate for an accurate benchmark of MLSFs. For example, many MLSFs for scoring are trained and tested on different PDBbind subsets (as described in Section 1.3.3), or use a test set consisting of protein-ligand complexes that were added to PDBbind after the complexes used in the training set, as a time split strategy[1,2]. However, recent studies have reported a decrease in performance when model training was based on careful partitioning of training and test sets based on ligand and protein similarities, when compared to training with a random, PDBbind-based, or time-based split—which indicate that performances of published models are often overoptimistic and affected by data leakage between training and test sets[3–8]. On the other hand, models trained exclusively on ligand features, protein features, or a combination of ligand and protein features without any information regarding their interactions showed comparable or better performance than models trained exclusively on interaction-based features, indicating a strong effect of ligand and protein memorization[5,9–11].

Ideally, the data split will consider both protein and ligand similarity. We initially implemented two similarity metrics to generate clusters of similar datapoints: the first based on the Bemis-Murcko scaffold[12] of ligands calculated using RDKit[13], and the second based on pocket 3D structural similarity using PocketMatch[14] program, following an approach similar to that of Kanakala et al[5]. However, combining the two types of clusters resulted in a single large cluster containing approximately 90% of the entire dataset, as ligand- and protein-based clusters had to be iteratively combined until no datapoint inside the combined cluster had any connection to the rest of the datapoints through shared cluster membership. It

was unrealistic to train with this split, due to the small size of the evaluation sets and the inability to perform cross-validation. Therefore, we used Bemis-Murcko scaffold approach to group protein-ligand complexes, in which molecules are simplified by removing monovalent atoms until only ring atoms and linker atoms remain[12]. Scaffolds of all ligands in the filtered 2P2IDB database were calculated from SMILES strings using RDKit[13], and ligands sharing the same scaffold were clustered together. Since our cross-docked datasets are composed of ligand poses docked into different proteins, each datapoint was simply grouped into the ligand's cluster.

Then, GNINA CrossDocked set was pruned to reduce the number of negative poses in each system, since it is quite uncommon to generate a maximum of 50 poses in VS, as we did in Section 2.2.2.3. The reduction was carried out by retaining the lowest root mean square deviation (RMSD) pose from each docked system, and randomly selecting 9 other poses from the pose pool. This down-sampling allowed us to create a training set which contains a typical number of poses generated from the docking scenario on prospective applications, at the same time including a diverse set of poses that comes from a more exhaustive pose sampling process (compared to generating ~10 poses with the docking program). One benefit of this method was that it reduced imbalance in the original dataset (**Fig. 2.4-B**), which can be problematic if the imbalance is too high.

Finally, the datasets were divided into fivefold clustered cross-validation (CCV) folds, in which the ligand-based clusters were iteratively assigned to each fold to prevent scaffold overlap between folds. CCV folds were also stratified to maintain the ratio between positive and negative samples as similar as possible in all five folds. The general principle of *k*-fold CV is as follows: (i) a dataset is split into *k* folds (ii) during each iteration, one fold is used as a validation set, while the model is trained on the rest of *k*-1 folds (iii) this process is repeated

until all folds were alternatively used as a validation set, and the performance is averaged across *k* folds. CCV was used in the training and validation of other recently developed MLSFs[3,5,7,8,15,16], as it is an effective way to measure performance on similarity-controlled "unseen" data, if an external test set is not used. As will be described in the subsequent sections, stratified CCV was utilized throughout model training, hyperparameter optimization, and model evaluation.

One concern with using the down-sampled dataset in CCV was that the sampling altered the distribution of positive and negative poses not only in the training folds but also in the validation folds. While down-sampling was a practical and necessary approach for managing the training data, the validation set should preserve the original class distribution to better reflect the data encountered during prospective use. We computed the proportion of positive poses in the down-sampled (lowest RMSD pose + 9 random poses) set, top-10 poses based on GNINA's CNNscore, and top-10 poses based on GNINA's minimizedAffinity*, with the assumption that during a prospective VS scenario, roughly 10 poses will be generated for rescoring (**Table 3.1**).

Table 3.1: The class distribution of GNINA CrossDocked set with different down-sampling techniques. The percentage of positive poses are roughly similar across 3 methods.

Sampling Method	Cutoff (Å)	Proportion of Positive Poses (%)
Top-10 CNNscore	3	12.88
	2	4.94
Top-10 minimizedAffinity	3	9.64
	2	3.91
Lowest RMSD pose + 9 Random	3	10.02
	2	4.84

* minimizedAffinity is a binding score from GNINA's semi-empirical SF, which is rescored by CNNscore.

With the RMSD <= 3 Å, the percentage of positives in the down-sampled set was 10.02 %, which is comparable to 12.88 % when top 10 poses were taken from the original pool of max 50 poses based on CNNscore, or 9.64 % based on minimizedAffinity. Since the distribution does not change significantly across different sampling techniques, we believe our down-sampling approach is justified and will have minimal effect on model evaluation.

3.2.2 Feature Generation

Over the past decades, countless methods have been developed to encode 3D molecular data into features, which are fed to machine learning models for the prediction of protein-ligand interactions. We applied the following criteria when selecting input features to encourage the MLSFs to learn interactions and prevent ligand and protein memorization. First, we chose 3D features when possible to prevent memorization of 1D/2D features. They also have less relevance to learning correct binding pose, as opposed to scoring or ranking scenarios. Second, we chose interaction features over atomic or molecular descriptors to force the model to learn interactions rather than isolate protein or ligand structures. Based on this, two types of 3D interaction features were selected: triplet interaction fingerprint (TIFP) and interaction graph, both of which were extracted using IChem program, a versatile toolkit with utilities for various structure-based computations[17].

To generate TIFPs, IChem was used to detect interacting protein and ligand atoms based on medicinal chemistry-informed pharmacophoric types and topological criteria, given a structure of the binding pocket. TIFPs encode the protein-ligand interactions into a vector of 210 integers, where each integer registers the count of unique triplets of detected atoms. Despite its one-dimensional format, TIFP is a robust input features, as it concisely encodes information about both topology and the pharmacophoric properties of protein and ligand atoms via triplet categorization[18]. TIFPs were calculated for GNINA CrossDocked sets by

running IChem *ints* module per the user guide (v.5.2.9)[17]. IChem rules for interaction detection and detailed TIFP generation process are shown in **Table 3.3** and **Fig. 3.7** in the appendix of this chapter, respectively.

Interaction graphs were generated from the same information that was used to calculate TIFPs. In addition to fingerprints, IChem *ints* module outputs a mol2 file containing atomic coordinates of interaction pseudoatoms (IPAs). In addition to the spatial information about IPAs, these mol2 files also contain each detected atom's interaction type, element, and identity (ligand/protein). Such mol2 files will be hereafter referred to as IPA_map. IPA_maps were generated for GNINA CrossDocked sets by running IChem *ints* module. Using in-house Python scripts, the IPA_maps were converted into graphs, where nodes represent either protein or ligand atoms. Edges were added between interacting protein–ligand atom pairs, as well as between protein–protein or ligand–ligand atom pairs within a 4 Å distance cutoff. Node features included a one-hot encoding of the interaction type and the node type (ligand or protein), while the sole edge feature was the Euclidean distance (Å) between connected nodes. The construction of this graph (hereafter referred to as *int-1*) was largely inspired by the work of Volkov et al.[10] To remain consistent with TIFPs, *int-1* graphs encode only spatial and interaction information related to the binding site of a docked ligand. However, due to the sparsity in the embedding space of these graphs, a second set of interaction graphs (hereafter referred to as *int-2*) was generated. *Int-2* graphs were constructed using the same IChem pipeline described above, with the exception that the detection radius for protein–ligand interactions was increased to 6 Å. The distance cutoff for protein–protein and ligand–ligand interactions remained at 4 Å. The node features were also augmented to include one-hot encoding of element type and buried surface area (BSA), which was calculated based on a given docked pose with dr-sasa program[19]. Along with solvent-accessible surface area

(SASA), BSA is an important property in PPI inhibitors as they tend to be more solvent-exposed than inhibitors of conventional protein-ligand complexes[20]. Not only is it an useful descriptor to discriminate between active and inactive molecules, SASA, and therefore BSA, varies with different docking poses[21], making it an ideal choice for our purposes. *Int-2* graphs contain much denser interatomic networks compared to *int-1* graphs and incorporate additional 1D and 3D atomic descriptors such as element type and BSA. To mimic the feature augmentation strategy adopted in *int-2* graphs, a second set of TIFPs were computed with an increased detection radius of 6 Å, but without incorporation of element types and BSA features. **Table 3.2** summarizes the node and edge features used in interaction graphs.

Table 3.2: Interaction graph node and edge features

Features	Type	Description	*Int-1*/*Int-2*
Interaction type	Node	One hot encoding of seven pharmacophoric properties: hydrophobic, aromatic, h-bond donor, h-bond-acceptor, positive ionizable, negative ionizable, metal.	Both
Node type	Node	One hot encoding of node type: ligand/protein atom.	Both
Element type	Node	One hot encoding of default RDKit atom types: C, N, O, F, P, S, Cl, Br, I.	*Int-2*
BSA	Node	BSA computed from the docked protein-ligand complex.	*Int-2*
Distance	Edge	Distance between two connected nodes in Å.	Both

3.2.3 Model Architectures

Random Forest (RF)

RF is a popular ensemble algorithm used in several different MLSFs (many of them listed in **Table 1.1**) for binding affinity prediction and virtual screening (VS) tasks. RF classifier makes predictions through majority voting among multiple decision trees and is

characterized by bootstrapping, random feature selection, and out-of-bag estimation techniques[22]. RF is a relatively simple, fast, and effective algorithm that can take TIFP input features and be a base of comparison for more complex architectures.

Multilayer Perceptron (MLP)

MLP, also known as a feed-forward neural network, is a type of neural network, consisting of an input layer, hidden layers, and an output layer. Each layer is composed of neurons with trainable weights and biases, which aggregate information and perform non-linear regression, allowing the model to learn complex patterns in the data. A loss function measures the error between predicted and true values. Using backpropagation, the gradients of this loss with respect to each parameter are computed, and gradient descent algorithm updates the weights to minimize the loss iteratively during training. MLP was employed with TIFP input features as another comparison with graph-based architectures.

Graph Neural Networks (GNNs)

This section outlines the framework of constructing a general GNN using the interaction graphs introduced in Section 3.2.2 for pose probability prediction. Given a graph $G = (V, E)$ where nodes $u_i \in V$ and edges $e_{ij} \in E$ represent a protein-ligand binding site, the GNN learns to distinguish between good and bad ligand poses. This is achieved by leveraging node and edge features, as well as the graph's topology, through a process known as message passing. In the first phase, a message containing information about neighboring node embeddings is generated by an aggregation function (**Eq. 3.1**):

$$m_{N(u)}^{(k)} = \text{AGGREGATE}\left(\{h_v^{(k)} | \, \forall v \in N(u)\}\right) \tag{3.1}$$

where $N(u)$ represents a set of neighbor nodes connected by an edge $e_{u,v}$, and $h_v^{(k)}$ a hidden

embedding at layer k†. The message at each node is updated by an update function to generate the hidden embedding for layer $k+1$ (**Eq. 3.2**).

$$h_u^{(k+1)} = \text{UPDATE}\left(h_u^{(k)}, m_{N(u)}^{(k)}\right) \tag{3.2}$$

To make a prediction on the graph-level, the final node embeddings must be combined to generate a graph embedding, which is known as graph pooling, or readout phase. Graph pooling is achieved by a readout function that maps a set of node embeddings to a graph embedding, h_G (**Eq. 3.3**).

$$h_G = \text{READOUT}(\{h_u | \forall u \in V\}) \tag{3.3}$$

Two different variations of GNNs were selected based on their initial predictive performance in a sample dataset, as explained as follows.

Graph Convolutional Network (GCN)

GCN is a type of GNN that employs symmetric-normalized aggregation and self-loop update, with message passing defined as such[23] (**Eq. 3.4, 3.5**):

$$m_{N(u)}^{(k)} = \sum_{v \in N(u) \cup \{u\}} \frac{h_v^{(k)}}{\sqrt{|N(u)|}\sqrt{|N(v)|}} \tag{3.4}$$

$$h_u^{(k+1)} = \sigma(W^{(k)} m_{N(u)}^{(k)}) \tag{3.5}$$

where $W^{(k)}$ is a learnable weight matrix and σ is an activation function. The GCN model was constructed by stacking multiple GCN blocks, each comprising a GCN layer, followed by batch normalization and a Rectified Linear Unit (ReLU) activation. After the final GCN block, a global mean pooling layer was applied to aggregate the final node embedding into a

† Note that at $k = 1$, h_v^1 is the input node embeddings.

single graph-level embedding. This graph embedding was then passed through a prediction block consisting of multiple fully connected (MLP) layers, each followed by batch normalization and ReLU activation. The final linear layer of the MLP used a sigmoid activation to produce a probability between 0 and 1. Dropout was applied at two separate occasions, once after readout and again after MLP prediction stack.

Graph Isomorphism Network (GIN)

GIN is a type of GNN that employs a sum aggregator and update function using MLP[24] (**Eq. 3.6**):

$$h_u^{(k+1)} = \mathrm{MLP}^{(k)}\left(\left(1+\epsilon^{(k)}\right)\cdot h_u^{(k)} + \sum_{v\in N(u)} \mathrm{ReLU}\left(h_v^{(k)} + e_{v,u}\right)\right) \tag{3.6}$$

where ϵ is a learnable parameter or a fixed scalar that adjusts the weight of a node relative to its neighbors. We use a variation of GIN which incorporates edge features, $e_{v,u}$, which was defined as the distance between two nodes[25]. The GIN model was constructed following the same architecture as the GCN model, with the GCNConv layers replaced by GINEConv layers from PyTorch Geometric[26] v. 2.5.3. The input graph structure, node and edge features, and both GNN architectures are shown in **Fig. 3.1**.

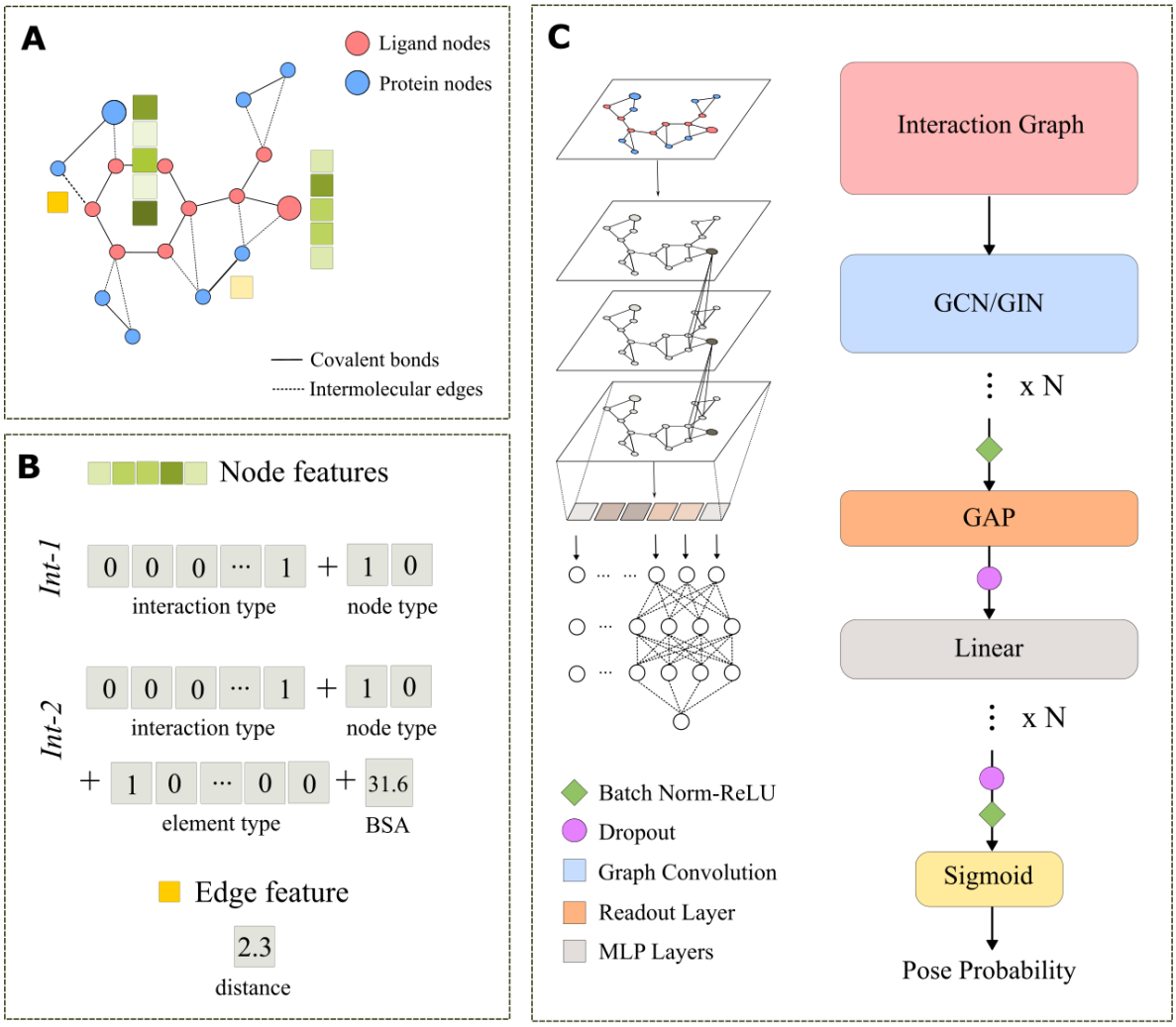


Figure 3.1: GNN architecture. **A**. Interaction graph is constructed from IChem detected protein and ligand nodes, with edges between covalent and intermolecular interactions. **B**. Node and edge features for *int-1* and *int-2*: *int-1* is composed of purely interaction-based features, while *int-2* is augmented with an atomic descriptor, element type, and a 3D descriptor, BSA. **C**. Architecture of two graph-based models, GCN and GIN. The models are composed of several graph convolution layers (GCN/GIN) followed by global average pooling (GAP) layer to create graph embeddings. The graph embeddings are trained by several MLP layers before outputting the final pose probability.

3.2.4 Model Training and Hyperparameter Optimization

Four models per architecture introduced in the previous section were developed using a combination of two RMSD cutoff values, 3 Å and 2 Å, and two input features, *int-1* and *int-2*[‡]. RF models were implemented using scikit-learn[27] v. 1.4.2, while DNNs (MLP, GCN, and

[‡] As mentioned in Section 3.2.2, two versions of TINFs were generated. The first version will hereafter be referred to as *int-1*, and the second version (with increased detection radius) as *int-2* (ex. RF-int1 and RF-int2) for the sake of consistency with *int-1* and *int-2* interaction graphs.

GIN) were implemented using PyTorch[28] v. 2.3.1. For RF and MLP, the fingerprints were standardized with Scikit-Learn's MaxAbsScaler[27]. Training was performed on fivefold CCV folds defined in Section 3.2.1, resuslting in five ensemble models each trained on 80% of the dataset. RF was trained with balanced class weight and bootstrapping enabled. The DNN models were trained with mini-batch gradient descent with a batch size of 100 to minimize the loss function shown in **Eq. 3.7**, which is a variant of the binary focal loss introduced by Lin et al[29].

$$Loss = -\alpha(1 - p_t)^{\gamma} log(p_t) \tag{3.7}$$

where p_t (**Eq. 3.8**) is defined as the probability of a pose having an RMSD less than a cutoff value, α is a weighting factor[§], and γ is a tunable focusing parameter.

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{otherwise} \end{cases} \tag{3.8}$$

Focal loss was used to address the imbalance in the datasets, in which the modulating factor, $(1 - p_t)^{\gamma}$, down-weighs the losses from well-classified examples ($p_t \to 1$) with increasing γ, pushing the models to focus on harder examples, i.e. false negatives. For MLP models, training was carried out for 50 epochs with an early stopping threshold of 10. For GNN models, training was carried out for 100 epochs with an early stopping threshold of 20.

Hyperparameter optimization (HPO) was performed using the Tree-structured Parzen Estimator search algorithm[30] via Optuna[31], wherein the parameters were tuned to maximize the mean of Best Pose success rates (defined in Section 2.3.2) across five CCV folds, calculated over systems found in each validation fold. Due to the resources required to perform a thorough search over hyperparameter space, only 8 out of 16 models presented in

[§] α parameter is the same as the one used in balanced cross entropy loss, which balances out positive and negative class. Normalized class weight of the training set was used as α in all models.

this work were optimized using Optuna[**]. The details regarding the explored hyperparameter space are presented in the Appendix of this chapter (**Table 3.4**).

3.2.5 Model Evaluation

The performance of models trained on the best set of hyperparameters was evaluated by fivefold CCV for the pose prediction task. During the final training loop, the fivefold splitting of the dataset was randomized by using a different random seed than what was used in HPO, in order to remove any potential learned bias during the tuning process. Receiver-operating characteristic (ROC) curves, introduced in Section 1.3.2.1, as well as their area under the curve (AUC) values, were computed and averaged over five CCV folds as a measure of the models' ability to differentiate between correct and incorrect ligand poses. As with HPO, the Best Pose success rates were calculated and averaged over five CCV folds to measure the models' ability to select the correct pose among the docked pose ensemble, which serves as a more meaningful metric in terms of potential real-world applications in docking and VS.

3.3 Results and Discussion

3.3.1 ROC-AUC

Docking power of trained models was evaluated with ROC curves for their ability to discriminate low-RMSD poses from high-RMSD poses defined by different cutoff values. As seen in **Fig. 3.2-A,B**, models trained on *int-1* features perform worse than GNINA, regardless of the input feature utilized. However, **Fig. 3.2-C,D** shows that models trained on *int-2* outperform GNINA if the features are interaction graph-based but fail to do so using TIFP

[**] For RF and MLP, only *int-1* models; for GCN and GIN models, only *int-2* models (both cutoff values). Best parameters found in each HPO was used in RF, MLP-*int2* models, and GCN/GIN-*int1* models, respectively.

features. It is worth noting that GIN models trained on *int-1* features are still nearly comparable with GNINA in terms of AUCs (3 Å: 0.86 vs. 0.88; 2 Å: 0.90 vs. 0.93) (**Fig. 3.3)**, despite being trained on minimalistic features and relatively sparse network—especially compared to how GNINA's input representation for training, which is a cube with a length of 24 Å to represent the binding site, using either atom type and Gaussian density functions as features[16].

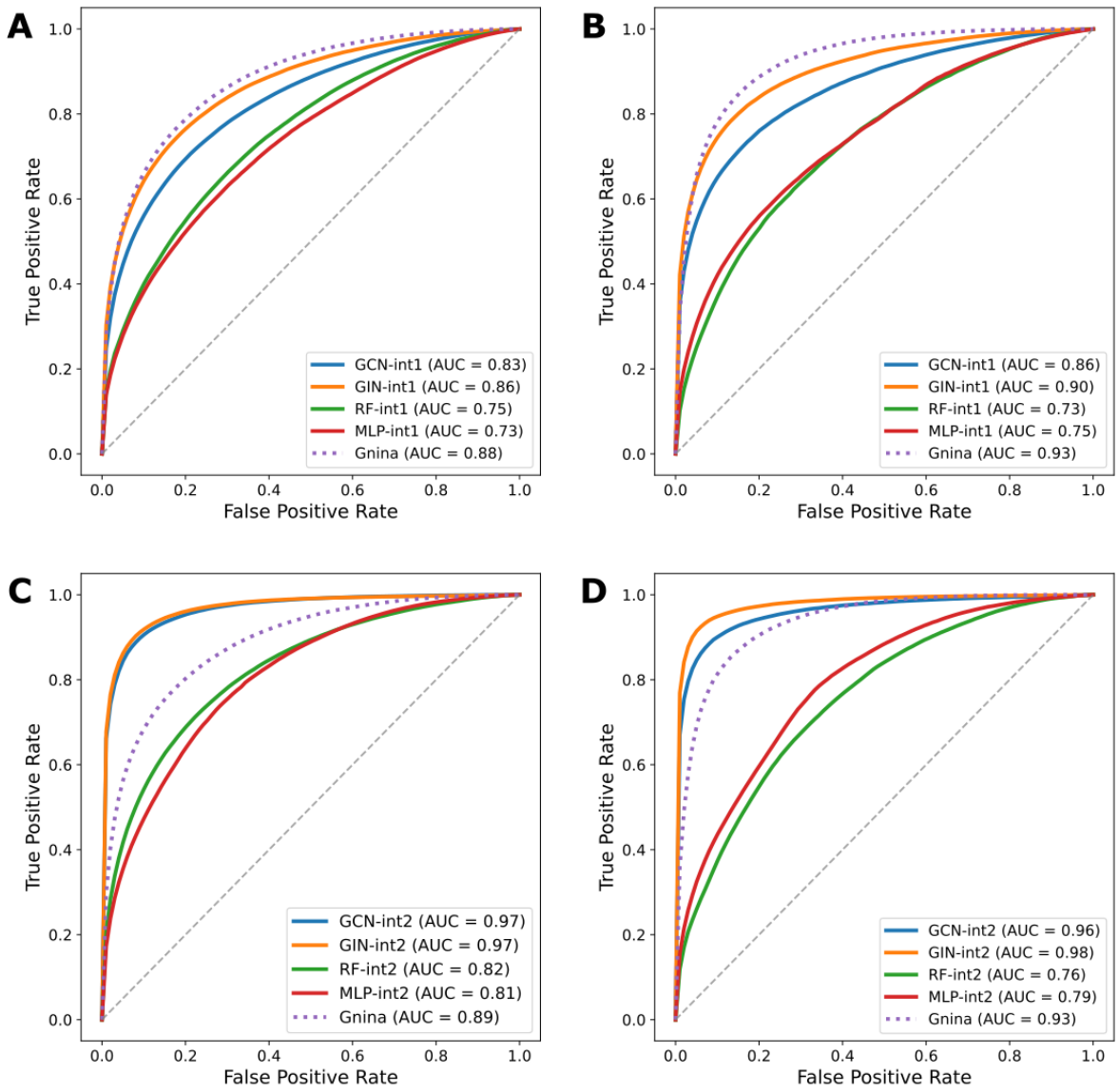


Figure 3.2: Model ROC curves averaged over fivefold CCV compared to GNINA. **A**. Models using RMSD cutoff of 3 Å and int-1 features. **B**. Models using RMSD cutoff of 2 Å and int-1 features. **C**. Models using RMSD cutoff of 3 Å and int-2 features. **D**. Models using RMSD cutoff of 2 Å and int-2 features.

As expected, using *int-2* features as opposed to *int-1* features improved the AUCs in all cases. The increase in AUCs observed from *int-1* to *int-2* was similar between using interaction graphs and TIFPs. For example, the AUC increase is on average around 0.1 between graph-based models going from *int-1* to *int-2*, and the maximum increase observed in RF and MLP models is 0.07 (RF 3 Å).

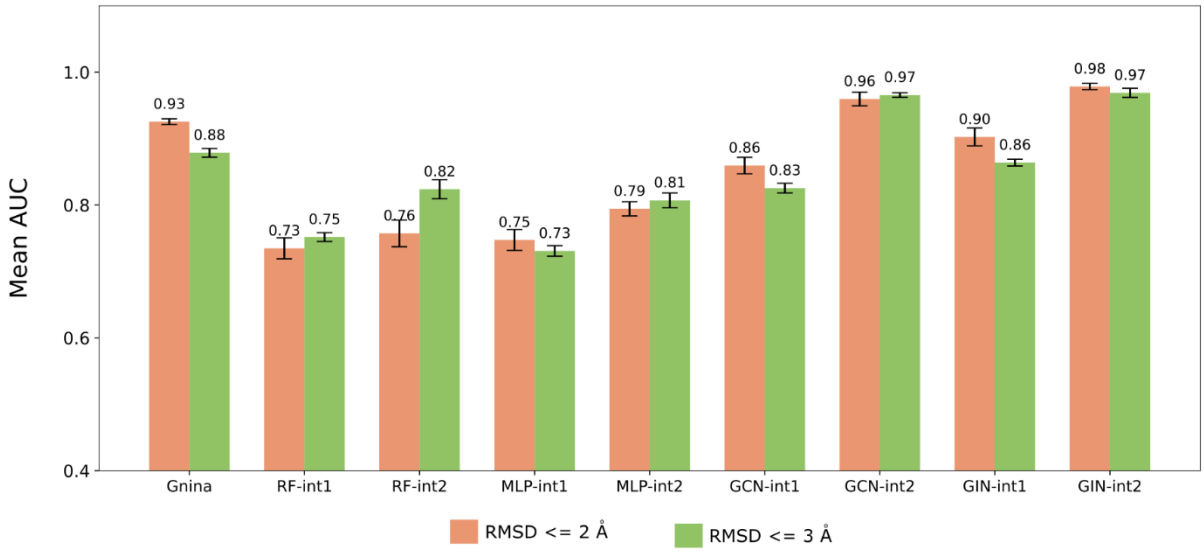


Figure 3.3: Model AUCs averaged over fivefold CCV compared to GNINA

Regarding the differences between 3 Å and 2 Å models, it was surprising to observe that half of the 2 Å models show higher AUCs than corresponding 3 Å models, considering the imbalance between positive and negative samples can more than double with lower RMSD cutoff. Given that GNINA evaluated with 2 Å shows worse AUC, it seems that the models are capable of handling skewed datasets.

3.3.2 Best Pose Success Rate

Next, docking power was evaluated with Best Pose success rate, which is the real testament to a model's ability to correctly identify near-native poses. Unlike AUCs, Best Pose success rates scale with respect to the All Poses success rate, which indicates the upper limit of docking power. Similar to our observations with ROC-AUCs, graph-based models with

int-2 features outperform GNINA in terms of Best Pose success rate. However, three of the four graph-based models trained with *int-1* features also outperform corresponding GNINA success rates, with the exception of GCN-int1 (**Fig. 3.4**). This supports our observation that GNNs, even when trained solely with interaction features can perform on par with GNINA in PPI database. On the other hand, there seems to be only a small benefit to embedding more information in TIFPs, since comparing *int-1* to *int-2* models show significantly lower improvement given the margin of error in success rate for RF and MLP models.

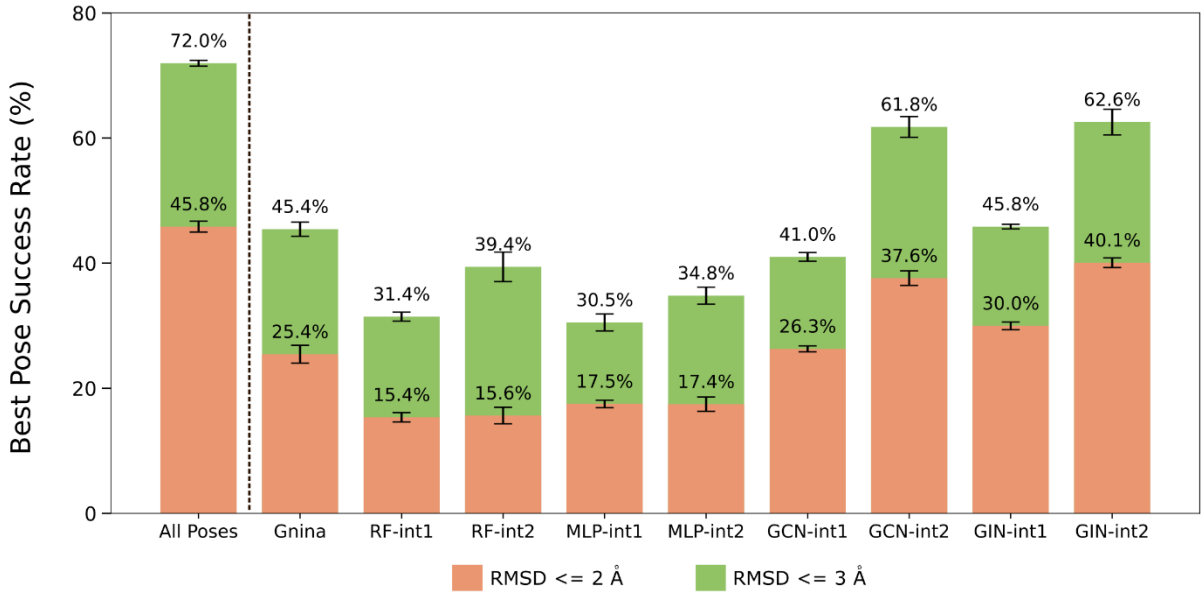


Figure 3.4: Model Best Pose success rates averaged over fivefold CCV in comparison with All Poses and GNINA success rates. Leftmost bar is not a Best Pose success rate, but All Poses success rate computed over the same CCV folds.

We have previously benchmarked the pose generation performance of GNINA in Chapter 2 by computing All Poses success rate, which is shown as the leftmost bar alongside the mode success rates on **Fig. 3.4**. Our best-performing model, GIN-int2, significantly bridges the gap between the Best Pose success rates of the baseline GNINA model and All Poses success rates, demonstrating the effectiveness of interaction-based 3D features and graph architecture. Given the high AUCs achieved by all graph-based models, the small gap that is still left to fill could be a result of the averaging of all systems in the validation folds, as some systems are

more challenging to determine the Best Pose than others—as it would be unrealistic to develop a scoring function that can generalize perfectly to any system.

3.3.3 The Effect of Down-Sampling in CCV Folds

In this section, we provide further justification of the down-sampling technique described in Section 3.2.1. To validate our models that simulate prospective VS, we removed any systems in the validation folds that do not occur in top 10 poses (CNNscore) in the calculation of success rates (**Fig. 3.5**). While this method has some limitations (notably, filtering out the poses with this approach removes roughly 7-8 poses from 10 in each system), our results showed that model performance did not change significantly when applied to more realistic data distribution (for example, 62.6 % vs. 58.7 % in GIN, *int-2*, 3 Å).

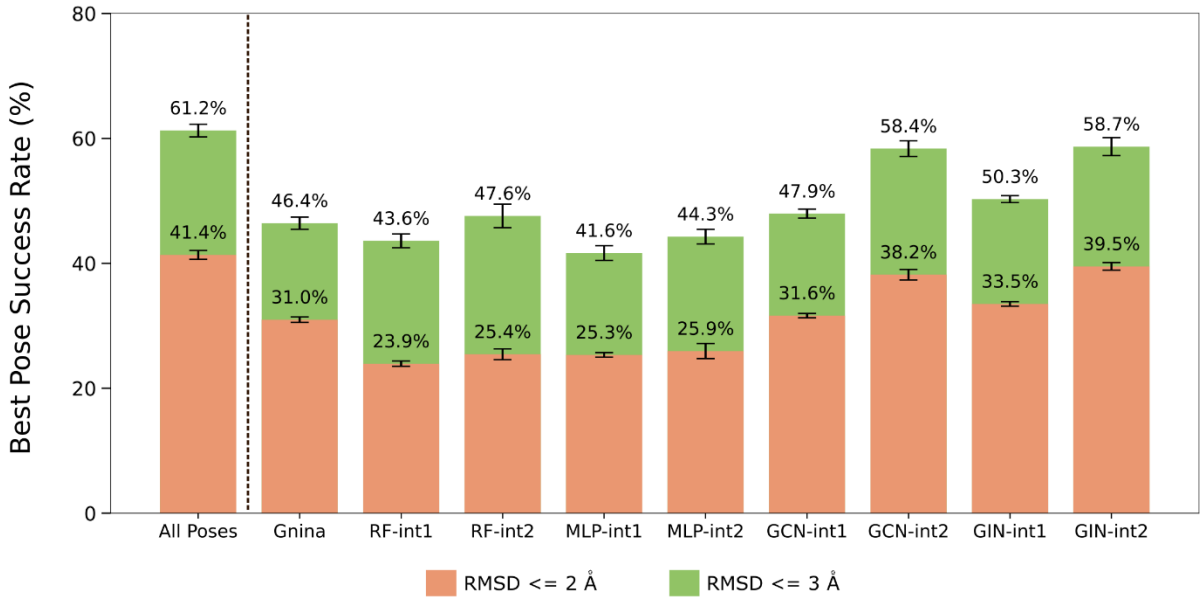


Figure 3.5: Best Pose success rates evaluated on validation folds, in which systems that are not part of top-10 (CNNscore) poses are filtered out. All Poses success rate was calculated by taking top 10 poses in the original pose database with a maximum of 50 poses.

3.3.4 Comparison with PIGNet2

Next, we compared our best models with another MLSF that was recently published. PIGNet2 is an MLSF for binding affinity prediction with a unique design, in which the model

learns the neighborhood of covalent bonds and intermolecular interactions via gated graph attention network and interaction network, respectively, and uses the final node features to parametrize physics-informed equations for energy terms[32]. We chose PIGNet2 not only because of its proven performance, which is superior or comparable to other state-of-the-art methods, but also because it is an interpretable model that can explain the strength and weakness of intermolecular atom-atom binding pairs, which makes it more robust against overoptimistic or biased results.

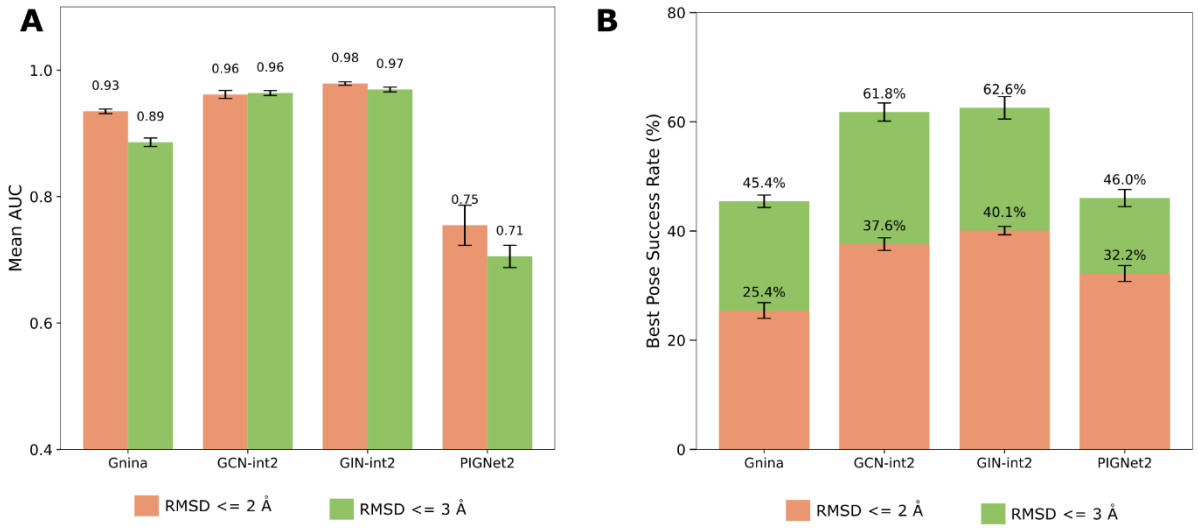


Figure 3.6: Comparison of different ML models developed in this work with PIGNet2, omitting overlapping data found in PIGNet2's training data. **A**. Mean AUCs **B**. Mean Best Pose success rates.

Since PIGNet2 was trained on PDBbind refined set, we removed any data from our CCV folds that occur in both datasets. PIGNet2's AUCs were computed by flipping the sign of the predicted affinities, as more negative binding affinity indicates stronger binding interactions. In terms of AUC, PIGNet2 seemingly performs much worse than GNINA **(Fig. 3.6-A**), while in terms of success rate, it slightly outperforms GNINA while underperforming compared to GCN and GIN models trained on *int-2* features (**Fig. 3.6-B**). The discrepancy between AUC and Best Pose success rate for PIGNet2 is most likely due to the fact that PIGNet2 predicts exact binding affinities (from the parametrized physics equations) rather

than pose probability. AUC is measured based on all poses in the system while success rate is measured only by the pose with the highest probability—since binding affinity, unlike probability, is not normalized between systems (e.g. best pose binding affinity for some systems could be higher than binding affinities for bad poses in another system), making this metric less useful in this case.

3.4 Conclusions

In this chapter, we developed MLSFs for the prediction of accurate binding poses for PPI inhibitors. To effectively capture the intermolecular interactions between protein and ligand molecules, we generated interaction graphs consisting of protein and ligand atom nodes and edges representing both covalent bonds and intermolecular interactions. Node and edge features were chosen to minimize learning ligand and/or protein structures by avoiding exclusively using 1D/2D atomic descriptors and instead utilizing 3D interaction features extracted from docked binding pockets. Despite its relatively sparse and simple representation and therefore requiring much less resources than training a CNN, graph-based input features trained with GCN and GIN were able to predict correct binding poses better than GNINA, as evidenced by higher Best Pose success rates during CCV. Our models also outperformed PIGNet2, a GNN-based MLSF that has been benchmarked against other state-of-the-art methods for pose prediction. By improving upon this metric, our MLSFs successfully bridged the gap between pose generation and identification of near-native poses on GNINA CrossDocked set. The best performing model, GIN, also demonstrated that it is possible to outperform GNINA by using purely interaction-based features, which is a further testament to the robustness of our methods in minimizing ligand and protein memorization issues. MLSFs presented in this chapter can be applied in VS campaigns against PPI target to identify near-native, low-RMSD poses for each docked system with a higher accuracy than

other generic MLSFs.

Despite the promising results, there are limitations in our approach, especially in model evaluation. First of all, we clustered our data based on ligand similarity, but pocket similarity was not taken into consideration due to the impracticality of jointly clustering to create CCV folds. While we believe that pocket memorization is minimal due to our choice in features and model architecture, pocket-based split models could be developed to better estimate this possibility. Secondly, ablation study can be performed by removing non-interaction-based features from *int-2*, for the same goal. Lastly, prospective validation could be performed by implementing our best performing models in an actual VS campaign and experimentally validating the activities of virtual hits.

3.5 References

(1) Stärk, H.; Ganea, O.-E.; Pattanaik, L.; Barzilay, R.; Jaakkola, T. EquiBind: Geometric Deep Learning for Drug Binding Structure Prediction. **2022**. https://doi.org/10.48550/ARXIV.2202.05146.

(2) Lu, J.; Hou, X.; Wang, C.; Zhang, Y. Incorporating Explicit Water Molecules and Ligand Conformation Stability in Machine-Learning Scoring Functions. *J. Chem. Inf. Model.* **2019**, *59* (11), 4540–45. https://doi.org/10.1021/acs.jcim.9b00645.

(3) Francoeur, P. G.; Masuda, T.; Sunseri, J.; Jia, A.; Iovanisci, R. B.; Snyder, I.; Koes, D. R. Three-Dimensional Convolutional Neural Networks and a Cross-Docked Data Set for Structure-Based Drug Design. *J. Chem. Inf. Model.* **2020**, *60* (9), 4200–4215. https://doi.org/10.1021/acs.jcim.0c00411.

(4) Li, J.; Guan, X.; Zhang, O.; Sun, K.; Wang, Y.; Bagni, D.; Head-Gordon, T. Leak Proof PDBBind: A Reorganized Dataset of Protein-Ligand Complexes for More Generalizable Binding Affinity Prediction. *ArXiv* **2023**, arXiv:2308.09639v1.

(5) Kanakala, G. C.; Aggarwal, R.; Nayar, D.; Priyakumar, U. D. Latent Biases in Machine Learning Models for Predicting Binding Affinities Using Popular Data Sets. *ACS Omega* **2023**, *8* (2), 2389–2397. https://doi.org/10.1021/acsomega.2c06781.

(6) Scantlebury, J.; Vost, L.; Carbery, A.; Hadfield, T. E.; Turnbull, O. M.; Brown, N.; Chenthamarakshan, V.; Das, P.; Grosjean, H.; Von Delft, F.; Deane, C. M. A Small Step Toward Generalizability: Training a Machine Learning Scoring Function for Structure-Based Virtual Screening. *J. Chem. Inf. Model.* **2023**, *63* (10), 2960–2974. https://doi.org/10.1021/acs.jcim.3c00322.

(7) Zhu, H.; Yang, J.; Huang, N. Assessment of the Generalization Abilities of Machine-Learning Scoring Functions for Structure-Based Virtual Screening. *J. Chem. Inf. Model.* **2022**, *62* (22), 5485–5502. https://doi.org/10.1021/acs.jcim.2c01149.

(8) Yang, J.; Shen, C.; Huang, N. Predicting or Pretending: Artificial Intelligence for Protein-Ligand Interactions Lack of Sufficiently Large and Unbiased Datasets. *Front. Pharmacol.* **2020**, *11*, 69. https://doi.org/10.3389/fphar.2020.00069.

(9) Morrone, J. A.; Weber, J. K.; Huynh, T.; Luo, H.; Cornell, W. D. Combining Docking Pose Rank and Structure with Deep Learning Improves Protein–Ligand Binding Mode Prediction over a Baseline Docking Approach. *J. Chem. Inf. Model.* **2020**, *60* (9), 4170–4179. https://doi.org/10.1021/acs.jcim.9b00927.

(10) Volkov, M.; Turk, J.-A.; Drizard, N.; Martin, N.; Hoffmann, B.; Gaston-Mathé, Y.; Rognan, D. On the Frustration to Predict Binding Affinities from Protein–Ligand Structures with Deep Neural Networks. *J. Med. Chem.* **2022**, *65* (11), 7946–7958. https://doi.org/10.1021/acs.jmedchem.2c00487.

(11) Durant, G.; Boyles, F.; Birchall, K.; Marsden, B.; Deane, C. M. Robustly Interrogating Machine Learning-Based Scoring Functions: What Are They Learning? November 2, 2023. https://doi.org/10.1101/2023.10.30.564251.

(12) Bemis, G. W.; Murcko, M. A. The Properties of Known Drugs. 1. Molecular Frameworks. *J. Med. Chem.* **1996**, *39* (15), 2887–2893. https://doi.org/10.1021/jm9602928.

(13) RDKit: Open-Source Cheminformatics. https://www.rdkit.org.

(14) Nagarajan, D.; Chandra, N. PocketMatch (Version 2.0): A Parallel Algorithm for the Detection of Structural Similarities between Protein Ligand Binding-Sites. In *2013 National Conference on Parallel Computing Technologies (PARCOMPTECH)*; IEEE: Bangalore, India, 2013; pp 1–6. https://doi.org/10.1109/ParCompTech.2013.6621397.

(15) Shen, C.; Hu, X.; Gao, J.; Zhang, X.; Zhong, H.; Wang, Z.; Xu, L.; Kang, Y.; Cao, D.; Hou, T. The Impact of Cross-Docked Poses on Performance of Machine Learning Classifier for Protein–Ligand Binding Pose Prediction. *J. Cheminformatics* **2021**, *13* (1), 81. https://doi.org/10.1186/s13321-021-00560-w.
(16) Ragoza, M.; Hochuli, J.; Idrobo, E.; Sunseri, J.; Koes, D. R. Protein–Ligand Scoring with Convolutional Neural Networks. *J. Chem. Inf. Model.* **2017**, *57* (4), 942–957. https://doi.org/10.1021/acs.jcim.6b00740.
(17) Da Silva, F.; Desaphy, J.; Rognan, D. IChem: A Versatile Toolkit for Detecting, Comparing, and Predicting Protein–Ligand Interactions. *ChemMedChem* **2018**, *13* (6), 507–510. https://doi.org/10.1002/cmdc.201700505.
(18) Desaphy, J.; Raimbaud, E.; Ducrot, P.; Rognan, D. Encoding Protein–Ligand Interaction Patterns in Fingerprints and Graphs. *J. Chem. Inf. Model.* **2013**, *53* (3), 623–637. https://doi.org/10.1021/ci300566n.
(19) Ribeiro, J.; Ríos-Vera, C.; Melo, F.; Schüller, A. Calculation of Accurate Interatomic Contact Surface Areas for the Quantitative Analysis of Non-Bonded Molecular Interactions. *Bioinformatics* **2019**, *35* (18), 3499–3501. https://doi.org/10.1093/bioinformatics/btz062.
(20) Trisciuzzi, D.; Nicolotti, O.; Miteva, M. A.; Villoutreix, B. O. Analysis of Solvent-Exposed and Buried Co-Crystallized Ligands: A Case Study to Support the Design of Novel Protein–Protein Interaction Inhibitors. *Drug Discov. Today* **2019**, *24* (2), 551–559. https://doi.org/10.1016/j.drudis.2018.11.013.
(21) Singh, N.; Chaput, L.; Villoutreix, B. O. Fast Rescoring Protocols to Improve the Performance of Structure-Based Virtual Screening Performed on Protein–Protein Interfaces. *J. Chem. Inf. Model.* **2020**, *60* (8), 3910–3934. https://doi.org/10.1021/acs.jcim.0c00545.
(22) Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45* (1), 5–32. https://doi.org/10.1023/A:1010933404324.
(23) Kipf, T. N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. arXiv 2016. https://doi.org/10.48550/ARXIV.1609.02907.
(24) Xu, K.; Hu, W.; Leskovec, J.; Jegelka, S. How Powerful Are Graph Neural Networks? arXiv 2018. https://doi.org/10.48550/ARXIV.1810.00826.
(25) Hu, W.; Liu, B.; Gomes, J.; Zitnik, M.; Liang, P.; Pande, V.; Leskovec, J. Strategies for Pre-Training Graph Neural Networks. arXiv 2019. https://doi.org/10.48550/ARXIV.1905.12265.
(26) Fey, M.; Lenssen, J. E. Fast Graph Representation Learning with PyTorch Geometric. arXiv 2019. https://doi.org/10.48550/ARXIV.1903.02428.
(27) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Müller, A.; Nothman, J.; Louppe, G.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, É. Scikit-Learn: Machine Learning in Python. **2012**. https://doi.org/10.48550/ARXIV.1201.0490.
(28) Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Köpf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; Chintala, S. PyTorch: An Imperative Style, High-Performance Deep Learning Library. arXiv 2019. https://doi.org/10.48550/ARXIV.1912.01703.
(29) Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. arXiv 2017. https://doi.org/10.48550/ARXIV.1708.02002.
(30) Watanabe, S. Tree-Structured Parzen Estimator: Understanding Its Algorithm

Components and Their Roles for Better Empirical Performance. arXiv 2023. https://doi.org/10.48550/ARXIV.2304.11127.
(31) Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A Next-Generation Hyperparameter Optimization Framework. arXiv 2019. https://doi.org/10.48550/ARXIV.1907.10902.
(32) Moon, S.; Hwang, S.-Y.; Lim, J.; Kim, W. Y. PIGNet2: A Versatile Deep Learning-Based Protein–Ligand Interaction Prediction Model for Binding Affinity Scoring and Virtual Screening. *Digit. Discov.* **2024**, *3* (2), 287–299. https://doi.org/10.1039/D3DD00149K.

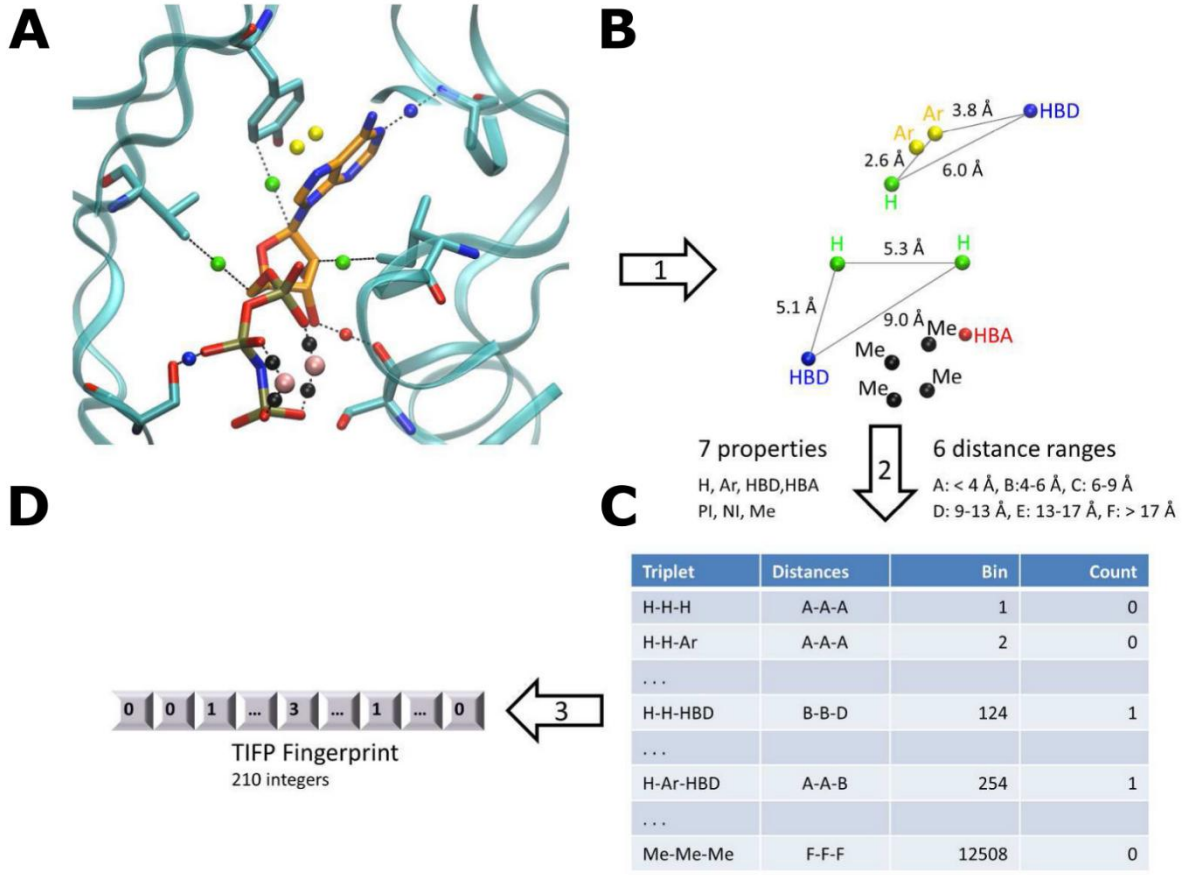
3.6 Appendix

Table 3.3: Default geometric rules for interaction detection, as defined by IChem. Reprinted from Volkov et al. "On the Frustration to Predict Binding Affinities from Protein–Ligand Structures with Deep Neural Networks," J. Med. Chem. **2022**, 65 (11), 7946–7958. Copyright 2022 American Chemical Society. Reprinted with permission.

Interaction	Rule 1[a]	Rule 2[b]
H-bond	$\left\|\overrightarrow{DA}\right\| \leq 3.5\mathring{A}$	$\left\langle \overrightarrow{DH}, \overrightarrow{HA} \right\rangle \in \left[\frac{-\pi}{4}, \frac{\pi}{4} \right]$
Ionic	$\left\|\overrightarrow{+-}\right\| \leq 4.0\mathring{A}$	
Hydrophobe	$\left\|\overrightarrow{Y_1Y_2}\right\| \leq 4.5\mathring{A}$	
Aromatic (Face to face)	$\left\|\overrightarrow{ac_1ac_2}\right\| \leq 4.0\mathring{A}$	$\left\langle \overrightarrow{n_1}, \overrightarrow{n_2} \right\rangle \in \left[\frac{-\pi}{6}, \frac{\pi}{6} \right]$
Aromatic (Edge to face)	$\left\|\overrightarrow{ac_1ac_2}\right\| \leq 4.0\mathring{A}$	$\left\langle \overrightarrow{n_1}, \overrightarrow{n_2} \right\rangle \in \left[\frac{\pi}{6}, \frac{5\pi}{6} \right]$
pi-cation	$\left\|\overrightarrow{ac+}\right\| \leq 4.0\mathring{A}$	$\left\langle \overrightarrow{n}, \overrightarrow{ac+} \right\rangle \in \left[\frac{-\pi}{6}, \frac{\pi}{6} \right]$
Metal	$\left\|\overrightarrow{MA}\right\| \leq 2.8\mathring{A}$	

[a]D: H-bond donor; A: H-bond acceptor; +: cation; -: anion; Y: hydrophobe; ac: geometric center of an aromatic ring; M: metal.

[b]H: Hydrogen; n: normal to the aromatic ring.



Triplet	Distances	Bin	Count
H-H-H	A-A-A	1	0
H-H-Ar	A-A-A	2	0
. . .			
H-H-HBD	B-B-D	124	1
. . .			
H-Ar-HBD	A-A-B	254	1
. . .			
Me-Me-Me	F-F-F	12508	0

Figure 3.7: IChem TIFP generation workflow. **A**. Given a binding site of a protein-ligand complex, IChem detects interactions on the fly. **B**. IPAs are characterized by seven pharmacophoric types (hydrophobic: H; aromatic: Ar; hydrogen-bond donor: HBD; hydrogen-bond acceptor: HBA; positive ionizable: PI; negative ionizable: NI; metal complexation: Me). **C**. All possible triplets of IPAs are generated and matched to a list. The count in each bin is encoded into a fingerprint of 12,508 integers. **D**. 12,508 integers are pruned into 210 integers by running IChem *ints* module with setting *--small -fgps STD*. Adapted from Desaphy et al. "Encoding Protein–Ligand Interaction Patterns in Fingerprints and Graphs," *J. Chem. Inf. Model.* **2013**, *53* (3), 623–637. Copyright 2013 American Chemical Society. Adapted with permission.

Table 3.4: Hyperparameter space explored in each model

ML Model	Hyperparameter Name	Description	HP Type	Explored Range
RF	Number of Estimators	number of decision trees	discrete	[200, 2000]
	Max Depth	maximum depth of each tree	discrete	[2, 64]
	Max Features	number of features to consider when looking for the best split	categorical	[sqrt, log2]
	Min Samples Split	minimum samples required to be at a leaf node	discrete	[2, 32]
	Bootstrap	whether bootstrap samples are used	binary	[True, False]
MLP	Linear Layers	number of linear layers	discrete	[1, 4]
	Hidden Channels	dimension of the hidden layers	discrete	[50, 200]
	Linear Dropout	dropout layer after each linear layer	continuous	(0.2, 0.5)
	Optimizer	optimizer function	categorical	[Adam, Adagrad, RMSProp]
	Learning Rate	model learning rate	continuous	(1e-5, 1e-3)
	Weight Decay	L2 regularization	continuous	(1e-6, 1e-3)
	Gamma (Focal Loss)	gamma value for focal loss	continuous	[0.5, 5]
GCN Stack	Graph Layers	number of GCN layers	discrete	[3, 4]
	Hidden Channels	dimension of the hidden layers	discrete	[64, 256]
	Graph Dropout	dropout rate for readout layer	continuous	(0.3, 0.6)
GIN Stack	Graph Layers	number of GIN layers	discrete	[2, 4]
	Hidden Channels	dimension of the hidden MLP layers	discrete	[64, 256]
	train_eps	trainable or set epsilon value (0)	categorical	[True, False]
	Graph Dropout	dropout rate for readout layer	continuous	(0.3, 0.6)
Prediction Stack	Prediction Layers	number of prediction layers after graph pooling	discrete	[2, 4]
	Hidden Channels	dimension of the hidden layers	discrete	[64, 256]
	Prediction Dropout	dropout layer after prediction layers	discrete	(0.3, 0.6)
GNN Training	Batch Size	size of batches during training	categorical	[128, 256, 512]
	Learning Rate	model learning rate	continuous	(1e-5, 1e-3)
	Weight Decay	L2 regularization for some optimizers	continuous	(1e-7, 1e-3)

	Gamma (Focal Loss)	gamma value for focal loss	discrete	[0, 5; 0.5]
	Optimizer	optimizer function	categorical	[Adam, AdamW, RMSProp]
	Node Latent Dimension	node dimension of the last graph convolution layer	discrete	[50, 100; 15]

Conclusions and Future Directions

This thesis outlines the design and development of machine learning scoring functions (MLSFs) to improve pose selection accuracy in molecular docking, in order to aid the structure-based drug design of small molecule inhibitors for protein-protein interaction (PPI) targets.

In Chapter 2, we benchmarked the docking power (pose prediction) of two docking programs, AutoDock and GNINA, on a database of PPI target-inhibitor complexes. While GNINA, which utilizes a more advanced sampling algorithm, successfully generated at least one near-native pose for 45.8% (or 71.6%, if the definition of near-native is expanded to include a greater margin) of the database, the benchmarking results revealed that the generic-purpose scoring functions (SFs) of both programs struggle to rank the near-native pose as a top pose, with the success rate of 22.3% and 38.0%, respectively. On one hand, the outcome of this work highlighted the need for PPI-specific MLSFs with enhanced docking power that can leverage the full potential of a given docking program's sampling algorithm. On the other hand, we developed several new databases consisting of poses of PPI inhibitors re-docked and cross-docked into binding pockets, which serve as valuable benchmarks for training and validating MLSFs for tasks related to pose prediction, such as pose classification, pose rank prediction, and pose root mean square deviation (RMSD) prediction. GNINA CrossDocked set, in particular, provides a large collection (6.2 million) of poses approximating realistic poses that may be encountered in docking and screening scenarios, making it particularly valuable for prospective applications in PPI drug discovery.

In Chapter 3, we trained and evaluated MLSFs with several different architectures and features extracted from the protein-ligand complex poses in GNINA CrossDocked set. Some

of the models trained on 3D structural interaction features and graph neural network (GNN) architectures exhibited competitive performance compared to state-of-the-art MLSFs for pose selection when evaluated on clustered cross-validation folds, providing some evidence that models are learning interaction patterns rather than relying on ligand memorization. Our best performing model achieved a success rate of 40.1% and 62.6%, substantially outperforming GNINA and PIGNet2 when benchmarked on the same dataset and successfully bridging the gap between the full potential of sampling algorithms and docking powers of existing SFs. In addition, we observed that embedding more information about the interactions between protein and ligand atoms in input features boosted the model performance, more so when the input representation was an interaction graph (for GNNs) than 1D fingerprint.

In summary, our work provides novel tools for small-molecule drug discovery for emerging PPI targets. We present new databases of PPI inhibitor poses, which serve as a valuable benchmark for the assessment of docking power in MLSFs. One caveat in our databases is that docked poses were generated using one program, which may render them less useful when evaluating an MLSF trained with poses from a different sampling algorithm. A potential solution is to combine AutoDock and GNINA datasets while ensuring poses from each system are sufficiently different by a tolerance of 2 Å (as has been done for each dataset), as well as further enlarge them by generating more poses using other popular docking programs.

Our interaction feature generation workflow and trained MLSFs serve as the basis for developing more accurate and robust models for PPI targets. The limitation in our methodology lies in model evaluation, in which the model performance was evaluated on cross-validation splits only based on ligand similarity. To reduce pocket memorization, as well as to better assess the impact of protein biases in our models, protein/pocket similarity

will need to be considered when splitting the dataset into cross-validation folds. However, the ultimate test for the effectiveness of our models will be in prospective validation, by performing virtual screening on a PPI target, followed by experimental confirmations of identified hits in binding assays.