Towards addressing structural data limitations in machine learning for small molecule drug discovery

Guy Durant

Linacre College

University of Oxford

A thesis submitted for the degree of

DPhil in Statistics

Trinity Term, 2025

Supervised by Dr. Fergus Boyles, Dr. Kristian Birchall, Prof. Brian Marsden and Prof. Charlotte M. Deane

Abstract

Drug discovery is an increasingly expensive and time-consuming process, with a drug taking over 10 years and $1.1 billion to bring to market on average. The recent rise of artificial intelligence and machine learning has promised to arrest and possibly reverse this trend. This thesis presents my work on analysing the impact of training data on these machine learning methods, specifically structure-based methods often used to enhance early-stage drug discovery and how I attempted to address their data's limitations.

In Chapter 2, I examine what machine learning-based scoring functions, designed to predict binding affinity for static protein-ligand complex structures, might actually be learning from their training data, specifically the widely-used PDBBind dataset. I compared the models to baseline models that obfuscated understanding of the underlying physics through their featurisation and found that these baseline models outperformed or equalled the performance of these advanced machine learning methods on the current and our proposed benchmarks. The results of this chapter suggest that the more likely source of improvement is unlikely to be the algorithm choice and instead could be found in changing the data that the scoring functions are trained on.

Another key task in structure-based drug discovery is the ability to accurately and rapidly predict the correct pose of a ligand when bound to a protein, a task known as docking. Typically, multiple poses are generated and then ranked or classified to get poses for further analysis. Chapter 3 focuses on the data needed to train a pose classifier, PoseTriager, that can be applied to several different docking software at once, enabling overall greater docking accuracy. I examine the impact of training simple graph neural networks on poses generated from a specific docking software, Smina, and how this impacts robustness to scoring noisy,

out-of-distribution poses. By training on these noisy poses and using data generated using my PoseFoundry pipeline, robustness is improved. This robustness does not correlate with stronger generalisation to scoring out-of-distribution poses from other software and into noisier protein structures, except in one case.

Increasing the amount of available structural data for training these structural methods, such as docking or scoring functions, could be a path to improved performance. In Chapter 4, I explore the capabilities of ligand-conditioned protein pocket design methods to generate plausible ligand pockets for synthetic structural data. As part of this, I developed additional validation and benchmarks to help drive the improvement of these methods. Generating physically plausible protein-ligand complexes is a major challenge for these methods, and these methods appear to have overfit to recapitulating the original sequence and do not explore pocket sequence space.

Finally, in Chapter 5, I began to implement a protein "trimming" and guidance method to speed up the structure prediction of ligand pockets using Boltz-1x, potentially enabling faster prediction of physically plausible structures, called ScrewzFix. Furthermore, I leveraged this faster inference to develop a hallucination method, Sparkz, that was able to optimise sequences for a given confidence metric. Further improvements to the techniques are needed, but they show promise in addressing the problems raised in Chapter 4.

Overall, this thesis highlights how the availability and nature of training data fundamentally shape the performance and limitations of machine learning methods in structure-based drug discovery, and explores avenues for generating new data to overcome these bottlenecks.

Acknowledgements

DPhils are challenging but incredibly rewarding experiences for a person, but they cannot be done alone. Success, as in all parts of life, is as dependent on the people around you as it is on yourself. This is a dedication to those who have been part of this journey with me.

First, I would like to thank my supervisors: Fergus, Kris, Brian and Charlotte, you have all been incredibly helpful and supportive. I am grateful for all the work, time and energy you all spent throughout getting me through the DPhil.

Thank you to the OPIGlets for creating such a great environment to work in, I feel honoured to have spent so many years amongst such talented, passionate and inspiring people. I look forward to seeing the great things you will all achieve in the future. Special thanks to Nele, Dylan, Gemma and Kit, with whom I've shared this entire journey.

Outside of work, I am incredibly fortunate to have made some fantastic friends who have brightened up my life and made these past four years the best of my life. Playing AFL in Oxford has definitely been a highlight with such a welcoming and fun environment that made me feel at home almost straight away. Thank you to Hazel, Isobel, Emily and Anna for the silly dinners and even sillier laughs; Julian and Graeme, for the wacky experiences together and Howie, for being a great housemate and even better friend.

Ellen, your intelligence, work ethic and tenacity constantly inspire me, and I've loved our adventures, quiet evenings and goofy times together. Thank you for being there when I needed to vent, a pep talk or just to chat some nonsense.

Dad, Mum, Charlotte and Georgina, I love and appreciate you all. Your unwavering support has been my greatest strength. I look up to each of you with gratitude and admiration.

Contents

Abbreviations

ADMET	absorption, distribution, metabolism, excretion and toxicity
AI	artificial intelligence
AUPRC	Area under the Precision–Recall curve
AUROC	Area under the Receiver Operating Characteristic curve
BIRD	Biologically Interesting Molecule Reference Dictionary
CADD	computer-aided drug design
CCD	Chemical Component Directory
CNN	convolutional neural network
DDPM	denoising diffusion probabilistic model
DMSO	dimethyl sulfoxide
DMTA	design, make, test, analyse
EGNN	equivariant graph neural network
FAPE	frame aligned point error
FEP	free energy perturbation
GAN	generative adversarial network
GAP	global average pooling
GAT	graph attention network
GNN	graph neural network
IPA	invariant point attention

LigandMPNNSCPacker	LigandMPNN Side Chain Packer
MCC	Matthews Correlation Coefficient
MDP	Markov Decision Process
ML	machine learning
MLBSF	machine learning-based scoring function
MM/GBSA	Molecular Mechanics using Generalised Born Surface Area
MM/PBSA	Molecular Mechanics using Poisson-Boltzmann Surface Area
MSA	multiple sequence alignment
MSE	mean squared error
ODDT	Open Drug Discovery Toolkit
ODE	ordinary differential equation
OOD	out-of-distribution
PAIN	pan-assay interfering compound
PDB	Protein Data Bank
PROTACs	proteolysis-targeting chimaera
QSAR	quantitative structure-activity relationship
RMSD	root mean squared deviation
RMSE	root mean squared error
RNN	recurrent neural network
SAR	structure-activity relationships
SASA	solvent accessible surface area
SBDD	structure-based drug discovery
SDE	stochastic differential equation
SGD	stochastic gradient descent
SPR	surface plasmon resonance
VAE	variational auto-encoder

1 Introduction

This chapter is based on work described in the following publication:

Guy Durant, Fergus Boyles, Kristian Birchall, and Charlotte M. Deane (2024). *The future of machine learning for small-molecule drug discovery will be driven by data. Nature Computational Science*, 4(10):735–743.

1.1 Motivation

The development of therapeutics to treat medical conditions is both a major scientific and industrial pursuit and a morally essential endeavour. These therapeutics can come in many forms, with popular modalities including small synthesised chemicals, referred to as small molecules (Midlam 2020), immune proteins, such as antibodies (Chames et al. 2009), and vaccines, such as the COVID mRNA vaccines (Park et al. 2021). In 2021, the latest curation of this data, the World Health Organisation found that non-communicable diseases resulted in 63.4% of all mortalities and infectious diseases were the cause of 27.3% (World Health Organization 2024). This global health burden underscores the importance of developing methods that can aid this discovery and design process to help enable people to lead their chosen lives without unnecessary suffering. Some of these deaths are preventable through

unmet medical needs that novel therapeutics could address (Scavone et al. 2019). As of 2021, 500 treatments have been developed, but around 7000 human diseases have been identified (Austin 2021), underlying the need for new therapies.

In the infancy of the field of drug and pharmaceutical development, we discovered drugs somewhat serendipitously through experiments or through the repurposing of natural remedies (Ban 2006). However, with the development of other fields and technology, the field has adopted rational design and targeting of dysfunctional systems within the human body on the cellular or molecular level (Drews 2000). This approach requires a robust understanding of the target cellular processes and the prediction and testing of how a developed therapeutic might have positive and negative impacts on a patient. This challenging process limits the rate at which we can do therapeutic discovery, with the average cost of getting a drug to market being $1.1 billion and now takes over 10 years (Wouters et al. 2020). This expensiveness is only increasing, a trend coined “Eroom’s Law", as since 1950, the number of new drugs approved per billion US dollars has halved approximately every 9 years, despite this move towards targeted therapeutic development (Scannell et al. 2012).

One way to address this concerning decline in productivity is by developing new methodologies and processes that might help arrest and reverse this trend. The application of machine learning (ML) or artificial intelligence (AI) methods has become a focus in recent years for developing such methods for computer-aided drug design (CADD) (Tropsha et al. 2025). By learning from data measured and curated in historic experiments, models can be trained to make predictions or generate hypotheses that can be tested experimentally, potentially avoiding the “basic research and brute force bias” tendencies of pharmaceutical research highlighted in the original Eroom’s law publication (Scannell et al. 2012).

During the 2010s, ML techniques began to outperform other statistical methods, with the most notable example of this being the development of AlexNet, an end-to-end neural network for image classification, which achieved 41% higher accuracy than the next competitor (Krizhevsky et al. 2012). Other famous demonstrations of the power of this technology also began to arise, such as AlphaGo (Silver et al. 2016)

and GPT (Radford et al. 2018) outside of drug discovery. This sparked interest in developing methods for drug discovery, with early adopters achieving the best performances in the data challenges, such as the Merck molecular activity challenge (Ma et al. 2015) and Tox21 toxicity data (Huang et al. 2016). A landmark success was protein structure prediction, where DeepMind's AlphaFold2 won the CASP14 competition and effectively solved the long-standing "protein folding problem", the challenge of accurately predicting the 3D structure of proteins from their amino acid sequences (Jumper et al. 2021). This success continues to motivate the field to research how best to apply these ML and AI techniques to improve the efficiency of drug discovery.

However, one key limitation that holds back future success is not the ML or AI architectures themselves, but instead the data that is used to train them. All these grand successes in machine learning can be attributed to both the algorithm and the data availability. AlphaGo was able to learn from many iterations of self-play (Silver et al. 2016), and GPT and other large language models utilised the entire internet (Crawl 2021). Drug discovery is a massively high-dimensional and parameter optimisation problem that is expensive to explore using either simulations or experimentation, which limits the available data that can be used to train such AI and ML methods (Vamathevan et al. 2019).

One specific example of this that this thesis focuses on is the structural data available for small molecule protein-binding therapeutics. Structures of small molecule drugs bound to protein targets, and how tightly these small molecules bind, known as their binding affinity, have been determined by separate researchers but globally collated using techniques such as X-ray crystallography (Turnbull et al. 2013) and surface plasmon resonance (SPR) (Jecklin et al. 2009). This data has been curated into databases such as the Protein Data Bank (PDB) (Burley et al. 2017), PDBBind (Wang et al. 2005) and more recently, PLINDER (Durairaj et al. 2024), with dataset sizes from thousands to hundreds of thousands. However, it is unclear whether this quantity and its quality are sufficient to learn generalisable patterns in the data and whether the diversity of these structures sufficiently covers the space

needed for future therapeutic design efforts. Whether the research community can address this data paucity is an open research question and is a focus of this thesis.

In this chapter, I begin by outlining the drug discovery process for small molecules and the pitfalls. Then, I introduce the basics of machine learning and specific techniques that arise throughout the thesis. I outline the data problems in the drug discovery field and highlight specific examples, and focus on particular applications of structure-based ML methods for docking, binding affinity prediction and co-folding. Also, I highlight structural datasets used to train such models and the flaws and problems in the datasets. The chapter concludes by presenting the overall structure and key contributions of the thesis.

1.2 Small molecules drug discovery pipeline

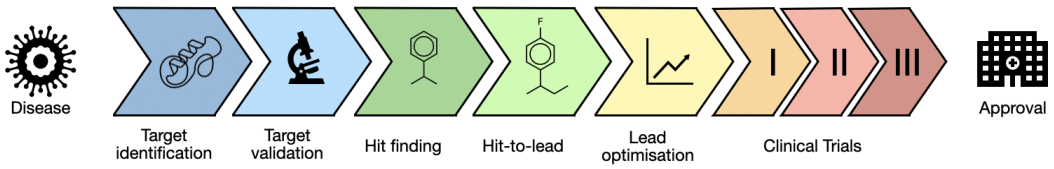


Figure 1.1: Overview of the small-molecule drug discovery pipeline, from disease identification, preclinical development, clinical trials, to regulatory approval.

1.2.1 Target Identification and validation

To develop a small molecule drug that can treat a disease, the intended target of that drug must be identified first from the vast number of proteins, RNAs and other macromolecules present in the cells of humans (Figure 1.1). This key step in the pipeline often does not differ from that of other modalities, such as antibodies and peptides. This identification typically involves a biological process or pathway that is aberrant in the disease. By developing a drug for a target within this pathway, the aim is to affect the path so that the drug prevents the disease phenotype or its impact on a patient is reduced (Hughes et al. 2011). To achieve this, two paradigms have been developed: target-driven drug discovery and phenotypic drug discovery. The

latter requires discovering a known compound that triggers the desired phenotypic change and is not the focus of this thesis (Vincent et al. 2022). Identifying a single or multiple targets for a potential therapeutic, target-driven drug discovery, does not often have a defined process (Emmerich et al. 2021). It can be a complex procedure of understanding the underlying mechanism of disease using various methods of biochemistry, molecular biology and bioinformatics, which typically comes from academic rather than industrial research (Emmerich et al. 2021). Examples of such methods include genome-wide association studies, which can identify mutations or genes associated with disease (Cerezo et al. 2025), and RNA interference or CRISPR-Cas9 gene editing, which can measure the impact of specific genes or their expression on disease pathology (Nidhi et al. 2021; Guidi et al. 2015).

Once a target within the mechanism has been identified, validation ensures that the target has potential therapeutic benefit and can have a small molecule drug developed against it (Figure 1.1). These checks are often crucial for the entire pipeline, as later failures can usually be linked back to poor target validation. For example, the attrition rate in phase II clinical trials could be reduced by 24% by better validation, which would subsequently lower the cost of development by 30% (Paul et al. 2010). Examples of validation required include the druggability of a target (Hughes et al. 2011), in the case of small molecules, would be the presence of a pocket to drug and potential safety issues of targeting (Gashaw et al. 2011). A key validation check is the establishment of a causal relationship between the target and the disease of interest (Cook et al. 2014). There are also other commercial concerns to consider, such as establishing the presence of any legal or intellectual property issues (Saha et al. 2011) or the unmet need within the patient population (Carlson et al. 2012), which determines the market for the future therapeutic.

1.2.2 Hit finding

Once a protein has been identified as a viable target for a small molecule, an initial compound, or hit, that binds to and modulates the protein, needs to be identified (Figure 1.1). If existing binders are known or exist in public databases

or patent literature, developing derivatives of these compounds can result in novel drugs that maintain or improve their efficacy (Zhao et al. 2009). This can be an efficient source of hits but heavily relies on the targets being well-understood, and so does not apply to novel targets (Ashraf et al. 2024). There are several well-established methods for doing so: high-throughput screening, fragment screening and target-specific de novo design.

High-throughput screening is the "brute force" testing of vast chemical libraries in their ability to bind to the protein (Carnero 2006). Typically, these libraries are not designed for specific targets in mind and cover chemical space sufficiently that the majority of targets can be hit (Villar et al. 2009). Binding can be identified using experimental assays, such as binding or cell-based assays (Macarrón et al. 2011), with chemical libraries or can be done virtually using virtual screening, using computational predictions (Badrinarayan et al. 2011). Despite the simplicity of the method, high false positives are common (Sink et al. 2010), and hits can be difficult to optimise when using experimental assays (Babaoglu et al. 2008). Furthermore, methodology relies on having a suitable assay applicable to it. If screening virtually, the ability to find hits is constrained by the accuracy of the function used to predict binding, which either uses physics-based functions, which can be insufficiently accurate (Klebe 2006) or ML methods, which suffer from a lack of generalisability to new chemical space (Guo et al. 2024).

Fragment-based drug discovery uses small libraries of low-molecular weight compounds (typically <300Da), named fragments, and assays the binding of these often promiscuous compounds (Kumar et al. 2012). By reducing the number of compounds assayed, the false positive and negative rates of high-throughput screening can be accounted for by prefiltering fragments and further confirmation of hits (Boettcher et al. 2010). Such assays include high-throughput structure determination by X-ray crystallography (Fearon et al. 2025) or biophysical assays such as SPR (Shepherd et al. 2022) and microscale thermophoresis (Linke et al. 2016). If structure determination is not used in the initial screen, promising fragments from the biophysical screens can still have their bound structures determined. By

combining these fragments, using fragment merging, linking or growing, novel hits that retain the interactions of the original separate fragments can be developed (Murray et al. 2012).

Target-specific de novo design is a more modern approach to hit finding, where initial hits are instead designed given a specific pocket of a target. ML and AI techniques have been applied to this task with some success in generating novel binders (Zhavoronkov et al. 2019; Blaschke et al. 2020). This method has the potential of exploring broader regions of chemical space in contrast to fragment-based drug discovery and high-throughput screening, which both are limited by the chemical libraries used as part of the initial screen (Lu et al. 2022a). However, these methods suffer from generating physically infeasible and synthetically intractable molecules, often requiring extensive filtering before proposed hits can be experimentally examined (Gao et al. 2020; Walters 2024). High-throughput screening, fragment-based drug discovery, and newer approaches such as de novo design are not mutually exclusive paradigms, but can be combined within the same drug discovery campaign to provide complementary sources of chemical starting points.

1.2.3 Hit-to-lead and lead optimisation

Once a hit or hits have been identified that demonstrate binding to the protein target of interest, the hits need to be prioritised for follow-up as potential drugs or leads, a process referred to as hit-to-lead (Figure 1.1). This prioritisation involves measuring or predicting properties including binding affinity, selectivity, and absorption, distribution, metabolism, excretion and toxicity (ADMET) (Hughes et al. 2011; Daoud et al. 2021). These ADMET properties can be assayed based on microsomal/hepatocyte clearance (Brian Houston et al. 1997) and membrane permeability (Menichetti et al. 2019), depending on the desired target. The balance of properties to optimise depends heavily on the target, for example, selectivity is weighted higher than potency for protein kinases due to the similarity within the protein family and so increased risk of off-target binding (Morphy 2010).

Furthermore, patentability and chemical tractability will be examined heavily at this stage, with deficiencies here detrimental to downstream clinical success (Keserű et al. 2006).

The next step, lead optimisation, aims to address deficiencies identified in the lead whilst maintaining existing favourable properties by making small chemical modifications (Hughes et al. 2011) (Figure 1.1). Such modifications include hit fragmentation, where larger leads are broken down into smaller fragments, and fragment-based methods (see above) are applied to generate new leads. Another strategy is hit evolution, in which different substitutions are tested to identify improvements and strengthen the structure-activity relationships (SAR). Finally, bioisosteric replacement involves substituting functional groups with alternatives that possess similar biological, physical, and chemical properties, while aiming to enhance features such as bioavailability or reduce toxicity (Keserű et al. 2006).

This complex multi-parameter optimisation requires extensive and expensive testing by a dedicated medicinal chemistry team, in a design, make, test, analyse (DMTA) cycle that informs future optimisation attempts (Andersson et al. 2009). Once optimal properties have been established, preclinical tests ascertain the potential of leads to transition through clinical trials successfully (Zhang et al. 2012). These tests typically focus on safety, toxicity and the pharmacokinetic and pharmacodynamic properties. Testing is generally done on animal models to represent the effects of the drug on the human body; however, due to ethical concerns, the use of animals is kept to a minimum, and there is hope that computer simulations and stem-cell organoid testing may also be used in the future to complement these tests (Passini et al. 2017; Park et al. 2024). The results of preclinical testing can also be used to inform the next DMTA iteration (Keserű et al. 2006).

1.2.4 Clinical trials and into the clinic

Once a lead compound has been designed, tested, and there is strong evidence that it is safe and effective for humans, the developers can now enter the designed drug into clinical trials. Clinical trials are split into three successive phases: Phase I,

Phase II and Phase III (Figure 1.1). Phase I trials aim to establish the safety of a drug, typically on a small number (20-80) of healthy volunteers (U.S. Food and Drug Administration 2018). Clinical trial investigators can utilise placebos, but typically not randomisation or control groups (Umscheid et al. 2011). Therefore, this phase does not gauge the efficacy of the drug, but increasing doses can be administered to establish the maximum dose that does not cause harmful side effects (Umscheid et al. 2011). The timing of new treatment and the best way to give the treatment, such as intravenously or orally, can also be determined. This process can take several months to complete (Umscheid et al. 2011).

The purpose of Phase II trials is to get an understanding of the effectiveness of the drug on the targeted disease. It requires 100-300 volunteers with the chosen disease for treatment and can take between several months and two years (U.S. Food and Drug Administration 2018). Due to the larger testing population compared to Phase I, further compiling of side effects and dosing can be done (Umscheid et al. 2011). The testing is often randomised and double-blinded to control for the effect of receiving treatment, with the control given either a placebo or an approved drug for the disease (Ildstad et al. 2001). Despite the increase, the testing population is still limited and so lacks statistical power. Instead, Phase II serves to aid in the planning of the Phase III trials by offering preliminary evidence rather than definitively proving efficacy (Umscheid et al. 2011).

If the preliminary data from Phase II is sufficient to justify the progression of the potential therapeutic, Phase III trials are started. Using much larger and more diverse populations (300-3000), these trials aim to confirm the efficacy of the treatment and safety profile (U.S. Food and Drug Administration 2018). Often several trials are needed as the statistical power is only high enough to find rare adverse effects that have a 1% chance (Onakpoya 2018). Upon passing Phase III, the developers of the drug can apply for marketing authorisation approval from regulatory bodies such as the FDA in the US (U.S. Food and Drug Administration 2020) and MHRA in the UK (Medicines et al. 2021). After this step, the developers can sell the therapeutic to treat a specific disease in the clinic. Even after the

approval, the side effects and safety of the drug continue to be monitored by regulators and the developers of the drug, often referred to as Phase IV studies (Umscheid et al. 2011).

Overall, this section has outlined a general pipeline for the development of a small molecule drug for a target. Each pipeline does not necessarily strictly follow these steps, depending on the particular disease. For example, if a drug is being repurposed and so its safety profile is well understood, it is possible to skip Phase I and II entirely (Cha et al. 2018). Successfully navigating this pipeline is incredibly difficult and is fraught with failure. In the next section, I explore some of the key challenges in developing small molecule therapeutics and what strategies might help overcome them.

1.2.5 Challenges in the pipeline

One of the first challenges in small molecule drug discovery is identifying a pocket to target successfully. Of the 19,370 protein-coding genes, there are considered to be only 3000 proteins in the "druggable genome" based on the presence of pockets that enable small molecules to alter function (Smith et al. 2024). The Target Central Resource Database lists only 1930 human proteins that have known small molecule binders (Sheils et al. 2021). To address this gap, Target 2035, a global initiative led by the Structural Genomics Consortium, aims to develop binders for all known proteins (Carter et al. 2019). Achieving this goal would require identifying hits for roughly 1,000 new protein targets each year, which presents a formidable challenge. Furthermore, often protein targets lack clear pockets to target, deemed "undruggable", yet frequently are the major drivers of disease (Zhang et al. 2024a). Some of these "undruggable" pockets have successfully been targeted using alternative modalities such as covalent inhibitors (Sutanto et al. 2020), molecular glues (Schreiber 2021) and proteolysis-targeting chimaera (PROTACs) (Li et al. 2022). A notable example of this is the KRAS G12C covalent inhibitor whose target was deemed undruggable due to its shallow and highly polar binding pocket

(Huang et al. 2021). Despite progress, many proteins remain undruggable to small molecules and so remains a major challenge.

Although small molecules are typically designed with a single target in mind, in practice, they frequently interact with multiple proteins. On average, a small-molecule drug binds 6–11 distinct targets beyond its intended one (Metz et al. 2010). These off-target interactions are a major contributor to drug attrition, with non-clinical toxicity representing the leading cause of failure (Ralston 2017). A clear example is hERG inhibition: nine structurally diverse drugs have been withdrawn from the market or faced labelling restrictions due to unintended activity against this potassium channel, a common off-target liability (Hishigaki et al. 2011). Despite the importance of such effects, pharmaceutical pipelines often limit off-target characterisation to secondary pharmacology screens covering only 11–104 proteins (Brennan et al. 2024), primarily due to time and resource constraints. More comprehensive profiling, however, is critical for improving the safety and success rates of small-molecule therapeutics.

The biggest challenge in small-molecule drug discovery, and for any modality, is successfully progressing through clinical trials. Between 2009 and 2018, the average failure rate across all modalities reached 84.6% (FTLOScience 2018). Despite advances in earlier stages of the pipeline, this high attrition rate has remained steady. For example, in the 1990s, poor drug-like properties, such as unfavourable pharmacokinetic profiles and poor bioavailability, accounted for 30–40% of failures, whereas in recent decades they have dropped to only 10–15% (Kola et al. 2004). Yet, this improvement has not translated into higher clinical success. Given that clinical development accounts for ∼60% of the total cost of drug discovery (Sun et al. 2022), this lack of impact is particularly concerning. Current estimates suggest that only 13.8% of therapeutics entering Phase I are ultimately approved, rising to 21.0% for Phase II and 59.0% for Phase III (Wong et al. 2019). Among the main reasons for failure, lack of clinical efficacy remains the dominant factor, responsible for 40–50% (Sun et al. 2022). Although preclinical tests in cell lines, tissues, and human disease models aim to mitigate this risk, the discrepancy between these

models and actual patient biology inevitably introduces uncertainty (Mahalmani et al. 2022). A "fail-fast" approach by improving the speed and reducing the cost of reaching clinical trials could help mitigate some of these inefficiencies (Porter 2023). ML and AI methods in computer-aided drug design could play a key role in enabling such approaches and accelerating and increasing the efficacy of the overall pipeline (Blanco-Gonzalez et al. 2023). The following section outlines the fundamental algorithms that underpin these ML and AI applications.

1.3 Machine learning

1.3.1 Supervised, unsupervised and reinforcement learning

ML algorithms learn patterns in data to perform tasks or make predictions without being explicitly programmed to do so (Mitchell 1997). AI is the capability of systems to perform tasks such as decision making, problem-solving and learning that are associated with human intelligence (Russell et al. 2021). Generally, the use of ML algorithms is what enables these capabilities; however, these terms are often used interchangeably.

This learning from patterns can be divided into three paradigms: supervised, unsupervised and reinforcement learning. However, these paradigms are not always distinct from each other, such as semi-supervised learning, a hybrid of supervised and unsupervised learning, and supervised-reinforcement learning.

Supervised learning is the fitting of a function

$$f_\theta : \mathcal{X} \to \mathcal{Y} \tag{1.3.1.1}$$

parameterized by θ, given a dataset of n samples $D = \{(x_i, y_i)\}_{i=1}^{n}$ where $x_i \in \mathcal{X}$ (inputs) and $y_i \in \mathcal{Y}$ (labels or targets). The learning occurs by minimising the expected loss (ℓ) or risk, which is typically an error between predictions and labels, such as mean squared error (MSE) or cross-entropy.

$$\theta^* = \arg\min_\theta \; \mathbb{E}_{(x,y)\sim P(data)}\Big[\, \ell(f_\theta(x),\, y) \,\Big] \tag{1.3.1.2}$$

As P(data), the probability distribution of the data, is unknown, the expectation of the empirical risk is instead approximated.

$$\theta^* = \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \ell\Big(f_\theta(x_i),\, y_i\Big) \tag{1.3.1.3}$$

This is the empirical risk minimisation principle that underpins supervised learning (Vapnik 1999). Examples of supervised learning are regression and classification. A basic case of this is simple linear regression, where the function takes the form.

$$f(x) = wx + b,$$

with w representing a single weight and b representing a single bias, the parameters. The parameters can be initialised with random values, and the predicted labels for those parameters can have their loss computed. The key step in enabling the loss to be minimised and accurate parameters to be found or updated (θ_{t+1}) relies on calculating the gradient of the loss:

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \left(\frac{1}{n} \sum_{i=1}^{n} \ell\Big(f_\theta(x_i),\, y_i\Big) \right). \tag{1.3.1.4}$$

However, computing the gradient across all datapoints at once can be expensive for large n; instead, subsets are taken, which introduces stochasticity into the optimisation of the parameters. This process is known as stochastic gradient descent (SGD) and is key for enabling training of functions with larger datasets (Robbins et al. 1951). In this example, however, a closed-form solution of the parameters of linear regression can be found, so using SGD for parameter optimisation here serves more as a simple demonstration of supervised machine learning rather than a practical one (Montgomery et al. 2021).

Unsupervised learning instead relies on fitting a function to a dataset of n samples with no labels, $\mathcal{D} = \{x_i\}_{i=1}^n$ with $x_i \in \mathcal{X}$ (inputs). Instead of outputting a label, the function aims to capture structure in the data (P_{data}) or identify common featurisations. To do this, the loss function measures how this structure is captured. Example of applications of unsupervised learning include clustering (Lloyd 1982), anomaly detection (Hodge et al. 2004) and dimensionality reduction (Hotelling 1933).

Finally, reinforcement learning aims to learn an optimal policy for an agent given an environment from a reward function. This can be formulated as the data being a Markov Decision Process (MDP):

$$D = (S, A, P, R) \tag{1.3.1.5}$$

where $\mathcal{S}$ is a set of states, $\mathcal{A}$ is a set of actions, P represents the transition probability to the next state for a given action and state (s, a), and R represents the reward received after taking a in s (Sutton et al. 1998). In classical reinforcement learning, policies and their rewards can be looked up; however, if the state/action space is large, machine learning models can be used to approximate the functions, known as deep reinforcement learning (Mnih et al. 2015). This thesis focuses specifically on supervised learning for small molecule drug discovery, where labels are the outputs of experiments. In the next section, I will outline the categories of machine learning algorithms that can be applied to any of the three paradigms outlined above. To start, I outline the basic principles of neural networks.

1.3.2 Neural networks

Neural networks are learnable functions inspired by the structure and function of biological neural networks in the brain (Rosenblatt 1958). Each neuron is connected to another with an associated weight and bias, and so can be considered separate simple linear regression models (see above) that take in the outputs of other linear regression models and feed their outputs to others (displayed in Figure 1.2). These nodes are typically organised into layers with a defined number of nodes per layer, with connections between all nodes in each layer, as shown in Figure 1.2. Increasing a network's depth or width increases its total number of trainable parameters (weights and biases). The number of layers and their widths are hyperparameters of the model and are picked before training. The first layer is usually termed the input layer and receives the input data, with each node taking a single feature or vector value. The last layer, the "output" layer, is what outputs the final value, whether

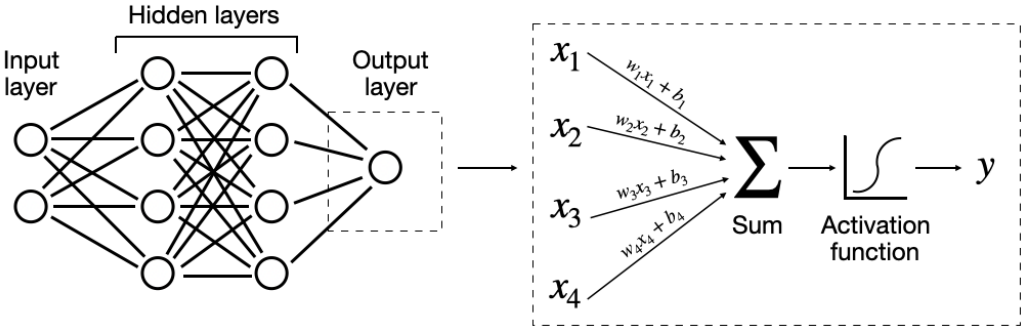


Figure 1.2: Schematic of a feedforward neural network. Left: a typical architecture with input, hidden, and output layers. Right: a single node, where inputs x_i are weighted, summed with a bias term, passed through an activation function, and output as y.

that be a classification probability or a regression prediction. The intermediate layers are called the "hidden layers" (Goodfellow et al. 2016).

SGD requires computing the gradient of the loss with respect to each parameter; updating parameters in the negative gradient direction can reduce the loss and enable learning. First, a forward pass is taken through the model to produce an output, from which the loss is calculated. To compute its gradient, a backward pass is taken through the model. At each layer, the gradient is propagated backwards by combining the upstream gradient with the local derivative of the layer's transformation. By applying this step recursively through the network, the full gradient for all parameters is obtained via the chain rule; this entire process is called backpropagation (Rumelhart et al. 1986).

Crucially, each node has a non-linear function applied to the sum of its inputs that allows the neural network to learn non-linear relationships. Without them, the neural network effectively collapses into a linear model. Examples of these non-linear functions or activation functions include ReLU, Sigmoid and tanh (shown in Figure 1.3) (Goodfellow et al. 2016). Due to this non-linearity, neural networks are theoretically universal approximators as they can learn to approximate any continuous function given sufficient training data (Hornik 1991). However, in practice, it is unknown what structure of neural network, such as the number of layers (depth) or the number of nodes in layers (width), will be the optimal approximator

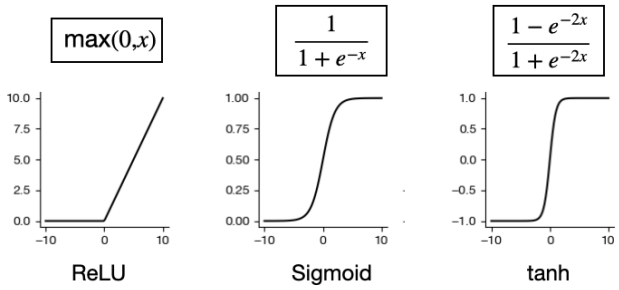


Figure 1.3: Examples of common activation functions used in neural networks. Left: Rectified Linear Unit (ReLU). Middle: Sigmoid. Right: Hyperbolic tangent (tanh). Each applies a non-linear transformation that enables neural networks to capture complex, non-linear relationships.

(Goodfellow et al. 2016). The structure of the network must be designed to allow the model to fulfil this capability, with inductive biases in these architectural choices guiding the model toward learnable and useful solutions (Goodfellow et al. 2016). Building on these foundations, I now outline three classes of neural networks particularly relevant to small-molecule drug discovery: convolutional neural networks (CNNs), graph neural networks (GNNs), and transformers.

1.3.3 Convolutional Neural Networks

CNNs, like the artificial neural network, are biologically inspired, but instead by how vision occurs in nature (Fukushima 1980). They consist of two specialised layers: convolutional and pooling, and are combined with fully connected layers (LeCun et al. 2002). Convolutional layers learn small kernels that detect local patterns such as edges or shapes to produce feature maps where those patterns occur. By taking the dot product of a restricted region of the input data or intermediate data with these kernels, an updated representation of the input data is produced. Sliding the kernel across the data transforms the entire input while only local features are considered at each step (see Figure 1.4). By capturing local and spatial connectivity in this way with the shared kernels, CNNs are translational equivariant, as the stack of feature maps shifts the same as inputs under translations (Cohen et al. 2016).

The pooling layers, again like the convolutional layers, operate on local regions of the input or feature maps, but instead of learning kernels, they apply fixed

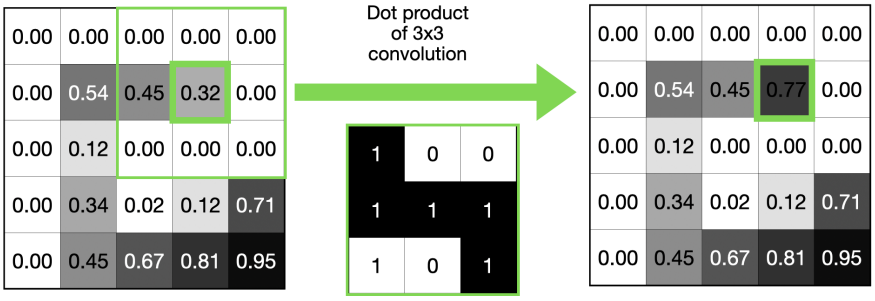


Figure 1.4: Schematic of a convolutional layer. Learnable kernels (filters) slide across the input to detect local patterns, producing feature maps that capture spatial connectivity.

functions that summarise these regions. For example, a 2×2 max-pooling layer replaces each 2×2 tile with its maximum value (see Figure 1.5). The output is downsampled and compressed by doing this across the entire feature map. The amount of downsampling depends on the stride, which is the step size with which the pooling window moves; for example, a stride of 2 means the 2×2 window shifts two pixels at a time (LeCun et al. 2002). Further, the development of skip

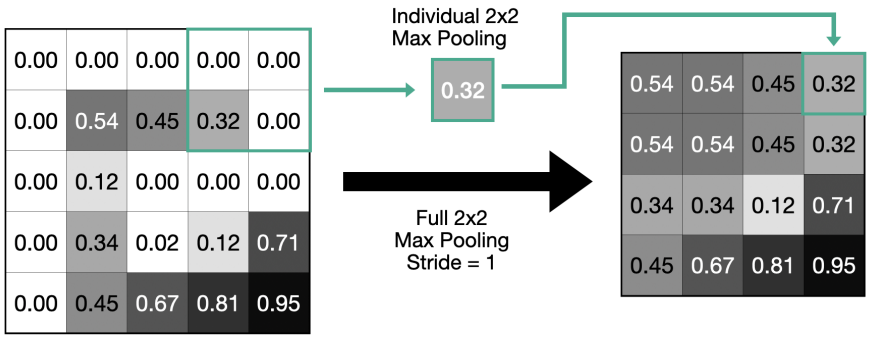


Figure 1.5: Schematic of a max pooling layer. Local regions (e.g., 2×2 tiles) are summarised by downsampling and compressing feature maps while retaining essential information.

connections enabled larger models without the "vanishing gradients" problem: for example, in ResNets, each residual block adds its input to the block's output (He et al. 2016), while in DenseNets, each layer receives as input the concatenation of all outputs from preceding layers (Huang et al. 2017).

They have achieved major success in computer vision (Krizhevsky et al. 2012), such as AlexNet, and have been applied to small molecule drug discovery (Wichard et al. 2015). By voxelising protein–ligand complexes, CNNs can predict molecular properties or even support generative tasks (Ragoza et al. 2017; Jiménez et al. 2018; Imrie et al. 2020). However, discretising atoms into voxels for convolution can distort fine-grained structural detail, motivating the growing popularity of GNNs (Qi et al. 2017).

1.3.4 Graph neural networks

GNNs are specialised neural networks that directly operate on input data structured as graphs. Graphs consist of nodes which are connected with edges and can model social networks (Kipf et al. 2016), recommendation systems (Ying et al. 2018), and knowledge graphs (Bordes et al. 2013). This structure mirrors the atomic structure of molecules and proteins with atoms and bonds, and so has been readily applied to learning from such structures (Gilmer et al. 2017). In doing so, GNNs learn their own feature representations directly from the graph, rather than relying on handcrafted descriptors.

Like CNNs, GNNs are composed of specialised layers, graph neural network layers, that operate directly on graphs. These layers are permutation equivariant: reordering the nodes in the input reorders the outputs (Kipf et al. 2016). This inductive bias is crucial for molecular applications, where the ordering of atoms must not affect the predicted properties (Gilmer et al. 2017).

A popular way to view GNNs is as message passing networks (Gilmer et al. 2017) that update node features based on the local neighbourhood of a node N_i, using "messages". A message function, MSG, computes these messages that takes in the central node (s_i) and a neighbouring node (s_j) that has an edge between them. A neural network typically parameterises the message function.

$$m_{ij} = \text{MSG}(s_i, s_j) \tag{1.3.4.1}$$

These messages are then combined using a permutation-invariance aggregation operator (AGG), such as the sum or mean. The node features are then updated

using this aggregated message by an update function, UPD, which can also be a learnt neural network.

$$s_i = \mathrm{UPD}\Big(s_i, \mathrm{AGG}_{j\in\mathcal{N}(i)}(m_{ij})\Big) \tag{1.3.4.2}$$

More sophisticated aggregation operators include learning an attention (described below) for each neighbour to create a weighted sum of the neighbours, first introduced in the graph attention network (GAT) (Veličković et al. 2017). This attention can also leverage edge features if included in the graph. The edge can also be updated using a different message function, MSG_E on the feature of the nodes it connects and its own features.

$$e_{ij} = \mathrm{MSG}_\mathrm{E}(e_{ij}, s_i, s_j) \tag{1.3.4.3}$$

This message-passing can be repeated for multiple iterations so that an increasing reach of the neighbourhood updates each node. This process is summarised in Figure 1.6. Finally, depending on the desired output type of the model, the graph features

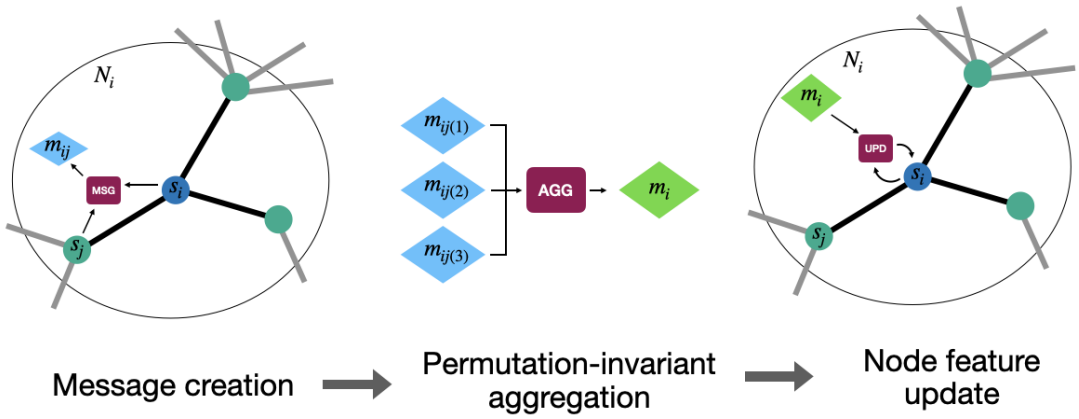


Figure 1.6: Illustration of the message passing framework in graph neural networks (GNNs). Messages are created between a node s_i and its neighbours s_j based on their features. A permutation-invariant aggregation function then combines the messages from the neighbours into a single message. This aggregated message is then used to update the node's feature representation.

of the final layer are mapped to a permutation-equivariant readout. If node or edge level information is required, a shared readout function can be learnt and applied

to each node. If a global graph property is being learnt, a graph pooling function is used to give a single vector for the whole graph. There exist many variants of GNN layers, differing in their aggregation functions and in how they combine node and edge features, leading to different strengths and weaknesses (Wu et al. 2020).

A limitation of standard graph neural networks is that they do not inherently respect physical symmetries such as SE(3). Incorporating these symmetries into the model constrains it to learn only functions consistent with the underlying physics, which is especially important for atomistic systems in three dimensions. Geometric GNNs were developed to address this limitation. In this thesis, I used a specific geometric GNN, the equivariant graph neural network (EGNN) for PoseTriager (in Chapter 3), which was also used by its predecessor, PointVS (Chapter 2) (Scantlebury et al. 2023). Here, I detail its implementation further.

This model represents atomic interactions using both Cartesian coordinates (vector-type features) and atom types (scalar-type features) as node features. The message is defined as

$$m_{ij} = \text{MSG}(e_{ij}, s_i, s_j, \|x_i - x_j\|^2), \tag{1.3.4.4}$$

where the squared distance between node coordinates, instead of the actual coordinates themselves, is also passed into the message function. This is key for maintaining equivariance in the network. These messages are then used to update the node features:

$$s_i = \text{UPD}(s_i, m_{ij}). \tag{1.3.4.5}$$

To update the coordinates, the authors define separate message and update functions:

$$x_i = x_i + \frac{1}{M-1}(x_i - x_j)\ \text{MSG}_{\text{coord}}(m_{ij}). \tag{1.3.4.6}$$

where M is the number of nodes in the neighbourhood.

1.3.5 Attention and transformers

Both CNNs and GNNs exploit locality, constraining interactions to nearby pixels or neighbouring nodes. In contrast, transformers discard this notion of strict locality: through the attention mechanism, every element in the input can attend to every other, allowing the model to learn patterns of global relevance.

Attention was first introduced to enhance recurrent neural network (RNN) by providing a larger effective context window, enabling them to capture long-range dependencies in sequences (Bahdanau et al. 2014). This concept was later generalised into the standalone self-attention mechanism, which no longer required the RNN and instead allowed each element of the input to attend to all others directly (Vaswani et al. 2017). Self-attention thus provides a flexible way for the model to weigh the relevance of different parts of the input when producing its predictions. This different weighting has clear applications in language where different words of a sentence have different importance depending on their context. The most common variant of self-attention is the scaled dot-product attention, which utilises three separate learnable weight matrices (W_q, W_k and W_v. The matrices project the inputs (x_i) into query, keys and value vector representations (q_i, k_i, v_i:

$$q_i = W_q x_i, \quad k_i = W_k x_i, \quad v_i = W_v x_i \tag{1.3.5.1}$$

The dot product between queries and keys provides a measure of their relevance, yielding unnormalised attention weights. To prevent excessively large values and improve numerical stability, these scores are scaled by $\sqrt{d_k}$. A softmax function then normalises the weights so they sum to one. Finally, these normalised weights are applied to the values, producing new query representations as weighted combinations of the values, with greater weight given to the most relevant inputs. The full attention equation is given here:

$$A(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V \tag{1.3.5.2}$$

Each set of weight matrices defines an attention head, and using multiple heads yields multi-head attention. Multi-head attention captures different types of

relationships in the data, and their outputs are concatenated to form a richer representation. Transformers use this self-attention as part of their architecture to build representations of the sequence data and have been applied to next-token prediction and translation tasks with significant success (Radford et al. 2015; Vaswani et al. 2017). For small molecule drug discovery, they have been applied for training models on large corpora of small molecules, for example, using tokenised SMILES strings (Honda et al. 2019; Chithrananda et al. 2020).

1.3.6 Decision tree-based models

A unique type of machine learning model that does not utilise neural networks is the decision-tree-based model. Decision trees are a popular model due to their simplicity for training, explainability and are often found to be highly accurate on tabular data (Breiman et al. 2017). I utilised tree-based models to train baseline machine learning scoring function models in Chapter 2, which I expand on in this section.

Each tree consists of nodes where comparisons using attributes or features of the given data point are made. Depending on the comparisons or "decision", the sample is passed to a specified connected node in a branching structure. The terminal node or leaf decides the value assigned to the sample, either a regression value or a classification. To train the decision tree, all features are considered, and thresholds are calculated that greedily reduce the "impurity" of the data. For classification, the impurity is measured using the Gini index at a node t and is defined as

$$G(t) = 1 - \sum_{k=1}^{K} p_k^2, \tag{1.3.6.1}$$

where p_k is the fraction of samples of class k in node t. For regression, the impurity is given by the MSE:

$$MSE(t) = \frac{1}{N_t} \sum_{i \in t} (y_i - \bar{y}_t)^2, \tag{1.3.6.2}$$

where N_t is the number of samples in node t and $\bar{y}_t$ their mean label. At each candidate split, the impurity reduction is computed as

$$\Delta I = I(t) - \frac{N_{t_L}}{N_t} I(t_L) - \frac{N_{t_R}}{N_t} I(t_R), \tag{1.3.6.3}$$

where $I(t)$ is the impurity at the parent node, and t_L and t_R are the left and right child nodes. The split that maximises ΔI is selected.

Model complexity is controlled by hyperparameters such as the number of features considered at each split, the maximum depth of the tree, or the fraction of data available at each node (Breiman et al. 2017). However, training can be computationally demanding on very large datasets, as all features and thresholds must be evaluated at each split, and models can become unwieldy or prone to overfitting (Breiman et al. 2017).

By combining multiple trees on subsets of the data, random forest models are trained that rely on the "wisdom of the crowd" principle that a majority vote is more accurate than a single model (Breiman 2001). This subsampling of data, known as bagging, reduces correlation between trees (Breiman 1996). By training the trees sequentially and learning the residuals or the gradient of the loss of the previous model with a decreasing learning rate or contribution to the overall prediction, more accurate models can be trained, such as XGBoost models (Chen et al. 2016). This learning of residuals is known as gradient boosting. These models are widely used for small-molecule property prediction based on descriptors or fingerprints, owing to their strong predictive performance and robustness (Kuz'min et al. 2011; Svetnik et al. 2003).

1.3.7 Generative modelling

One of the major advances in AI has been the development of generative models, which learn data distributions from finite samples and enable the generation of novel data consistent with those distributions (Ruthotto et al. 2021), such as images (Goodfellow et al. 2014), text (Radford et al. 2018) and video (Vondrick et al. 2016). The first algorithmic developments in the field include generative adversarial networks (GANs), which showed promise in image generation but showed instability in training (Goodfellow et al. 2014). GANs don't strictly learn the data distribution, but by learning to generate samples adversarially, they are compelled to generate data that is likely to come from the desired distribution. A limitation of this approach

is training instability, most prominently mode collapse, where the generator produces only a limited subset of possible outputs (Arjovsky et al. 2017). Another method proposed, variational auto-encoders (VAEs)), is trained to learn a consistent latent distribution that training data points can be encoded into and decoded out of without distorting accuracy (Kingma et al. 2013). VAEs had strong applicability to anomaly detection (Sun et al. 2018) and the learnt latent spaces are useful for downstream tasks (Xu et al. 2017); however, they have been shown to produce low quality or "blurry" data (Radford et al. 2015). More recently, transformers (Vaswani et al. 2017) (described above) have emerged as powerful generative models: by iteratively predicting the next token, pretrained models such as GPT (Radford et al. 2018) can generate coherent text. Nonetheless, this autoregressive approach is prone to compounding errors over long sequences, where early mistakes accumulate and reduce sample coherence and quality (Holtzman et al. 2019).

To overcome the limitations of the above methods, newer methods have been adopted, such as diffusion (Ho et al. 2020) and flow matching (Lipman et al. 2022). These methods learn ordinary differential equations (ODEs) or stochastic differential equations (SDEs) that map simple, known data distributions that are easy to sample from to complex data distributions that are otherwise difficult to sample from. This mapping is typically done iteratively by taking steps along the differential equations. These methods have shown remarkable success in image generation (Ho et al. 2020), de novo protein backbone generation (Watson et al. 2023) and even prediction tasks such as docking (Corso et al. 2022). Generative models are particularly powerful because they can represent complex conditional distributions and allow guided inference to steer outputs toward desired targets. In the following, I outline the principles of diffusion models and flow matching, together with conditioning and guidance strategies, as key background for Chapters 4 and 5.

Diffusion

Diffusion models, or denoising diffusion probabilistic models (DDPMs), are a class of generative models that learn to reverse a stochastic diffusion process, which

gradually transforms data into Gaussian noise (Ho et al. 2020). The forward process is a fixed Markov chain (equation shown below) in which Gaussian noise (ϵ) is sequentially added to the data according to a predefined variance schedule (β_t), until the original data distribution is transformed into a standard Gaussian.

$$x_t = \sqrt{1-\beta_t}\, x_{t-1} + \sqrt{\beta_t}\, \epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \quad \{\beta_t \in (0,1)\}_{t=1}^{T} \tag{1.3.7.1}$$

This can be expressed as each forward step being drawn from a Gaussian, which in turn enables a closed-form expression for $q(x_t \mid x_0)$ and direct sampling at any timestep for training

$$q(x_t \mid x_0) = \mathcal{N}\Big(\sqrt{\bar{\alpha}_t}\, x_0,\ (1-\bar{\alpha}_t)I\Big), \tag{1.3.7.2}$$

with $\alpha_t = 1 - \beta_t$, and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$.

The model is then trained to approximate the reverse process, step by step, denoising the Gaussian distribution back into samples from the data distribution by learning:

$$q(x_{t-1} \mid x_t) = \frac{q(x_t \mid x_{t-1})\, q(x_{t-1})}{q(x_t)}, \tag{1.3.7.3}$$

The marginal distributions $q(x_{t-1})$ and $q(x_t)$ require integration and so cannot be calculated in a closed form, necessitating the reverse process to be learnt.

However, the key insight that enables diffusion models to learn this reverse process is by keeping the time steps sufficiently small, which allows the reverse conditional distribution to be well-approximated as Gaussian. Therefore, the model does not need to learn an arbitrary distribution, but only the mean of the Gaussian, while the variance can be fixed according to the forward process. The reverse process can be parameterised as

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}\Big(x_{t-1};\ \mu_\theta(x_t, t),\ \Sigma_t\Big), \tag{1.3.7.4}$$

where Σ_t is the fixed variance. In practice, this is reparameterised in terms of the noise ϵ added during the forward process, yielding the simplified training objective (Ho et al. 2020):

$$\mathcal{L}_{\text{simple}}(\theta) = \mathbb{E}_{x_0,\epsilon,t}\Big[\|\epsilon - \epsilon_\theta(x_t, t)\|^2\Big], \tag{1.3.7.5}$$

where

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1 - \bar{\alpha}_t}\, \epsilon. \tag{1.3.7.6}$$

This reparameterisation reduces training to a denoising problem, where the model learns to predict the added Gaussian noise at each timestep. This is depicted in Figure 1.7. Alternative parameterisations are also possible: instead of predicting the noise ϵ, the network can be trained to predict the original clean sample x_0.

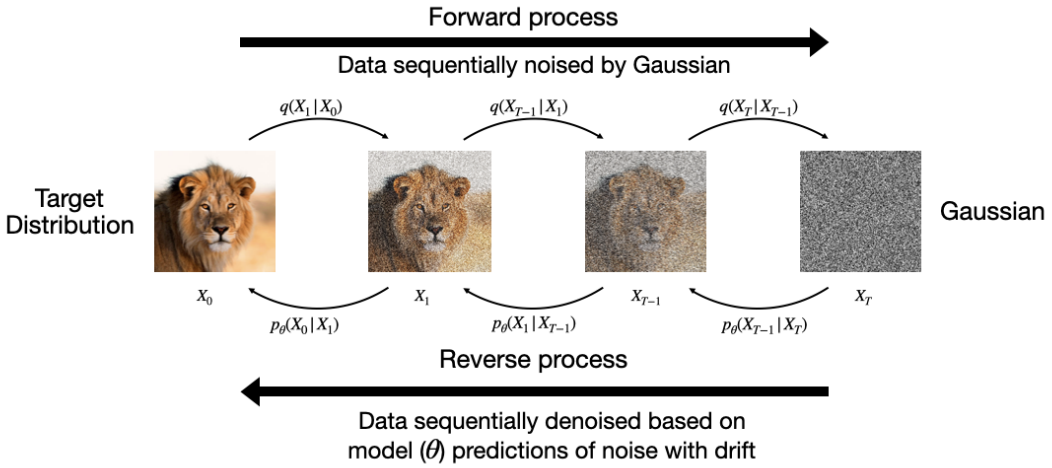


Figure 1.7: Illustration of a diffusion model. In the forward process, data x_0 is gradually noised into a Gaussian distribution x_T through successive steps. In the reverse process, a neural network learns to iteratively denoise x_T, reconstructing samples from the original data distribution.

To sample for the target distribution, a sample is taken from a Gaussian distribution, and the model predicts the noise expected to be added for that specific time step, iteratively denoising the data until the final time step is reached, transforming noise into the target data. This noise prediction can be considered as the model predicting the gradient of the differential equation or the score of the data distribution and taking steps along this score towards the target distribution (Song et al. 2020). By taking the mean of the predicted Gaussian, the model is solving an ODE, yielding a deterministic trajectory. At each time step, small amounts of noise are added to the sample within the predicted Gaussian noise removed (or denoised) ($\sqrt{\beta_t}z$), introducing variation within the prediction and

solving the SDE. An inference step is shown below:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \hat{\epsilon}_\theta(x_t, t) \right) + \sqrt{\beta_t} z, \tag{1.3.7.7}$$

The ability to model complex data distributions, together with flexible sampling, makes diffusion models particularly powerful for small-molecule drug discovery, where the 3D structures of molecules and proteins exhibit specific geometric details (Schneuing et al. 2024).

Flow matching

Flow matching can be viewed as a generalisation of diffusion models, providing a simpler framework in which models learn continuous flows that map arbitrary source distributions, beyond just Gaussians, into complex target distributions that are otherwise difficult to sample from (Lipman et al. 2022). By bypassing the need for the data to be mapped to Gaussian distributions in the forward process, it can be easily extended to learning data on manifolds (Chen et al. 2023). Furthermore, the mapping or paths it learns are more efficient than those sampled by diffusion models, leading to faster training and sampling (Lipman et al. 2022).

Unlike diffusion models, which iteratively predict noise to remove, flow matching models directly learn a vector field v_t that approximates the true field u_t, whose ODE trajectories transport the input distribution to the target distribution. Learning this vector field, known as the flow matching objective, directly is intractable, as infinite viable paths exist.

$$L_{\mathrm{FM}}(\theta) = \mathbb{E}_{t,\, p_t(x)} \left\| v_t(x) - u_t(x) \right\|^2 \tag{1.3.7.8}$$

Instead, the whole probability pathway is defined as a mixture of simpler probability paths, each conditioned on a single data point x_1. A simple interpolant between a sample from the base distribution and the target data point is often used to construct these conditional paths (shown in Figure 1.8). For continuous data, linear interpolation provides a natural choice, whereas for discrete data, specialised strategies are required since an interpolant is not straightforwardly defined.

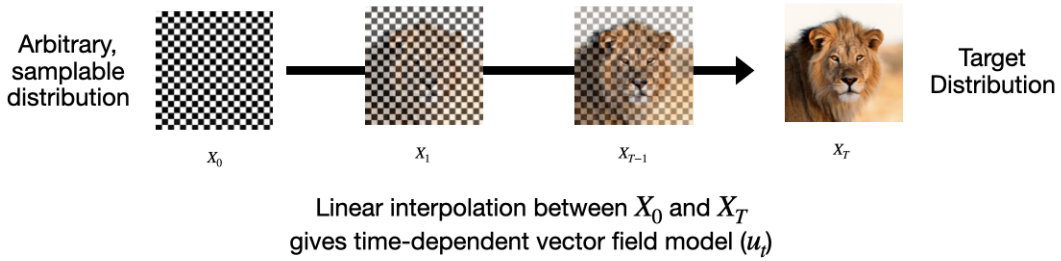


Figure 1.8: Illustration of flow matching. Instead of mapping data to Gaussian noise as in diffusion models, flow matching defines conditional probability paths between samples from an arbitrary source distribution and target data points. Linear interpolation between X_0 and X_T yields a time-dependent vector field u_t, which the model learns to approximate.

By marginalising the conditional probability paths (shown below), the marginal probability path p_t is obtained, which recovers the data distribution at $t = 1$.

$$p_t(x) = \int p_t(x \mid x_1)\, q(x_1)\, dx_1 \tag{1.3.7.9}$$

The global vector field can likewise be obtained by marginalising the conditional vector fields over the data distribution.

$$u_t(x) = \int u_t(x \mid x_1)\, q(x_1)\, dx_1 \tag{1.3.7.10}$$

The model learns the conditional flow matching objective, which shares the same gradients as the flow matching objective, and the two are monotonic functions of each other:

$$L_{\text{CFM}}(\theta) = \mathbb{E}_{t,\, q(x_1),\, p_t(x|x_1)} \left\| v_t(x) - u_t(x \mid x_1) \right\|^2 \tag{1.3.7.11}$$

Once trained, the model has learnt to accurately predict an approximation of the true vector field whose ODE can be solved to generate samples from the target data distribution. Its efficiency has made flow matching a popular alternative to diffusion models, such as in applications for 3D small molecule generation (Dunn et al. 2024).

Guidance and steering

Both diffusion and flow matching models, as described above, can be trained to sample from a target data distribution unconditionally. However, in many applications, it is desirable to sample from a specific region within this distribution,

for example, generating an image of a lion rather than any random image, or a protein conformation with particular structural features (depicted in Figure 1.9). To enable such control over the generative process, various guidance techniques

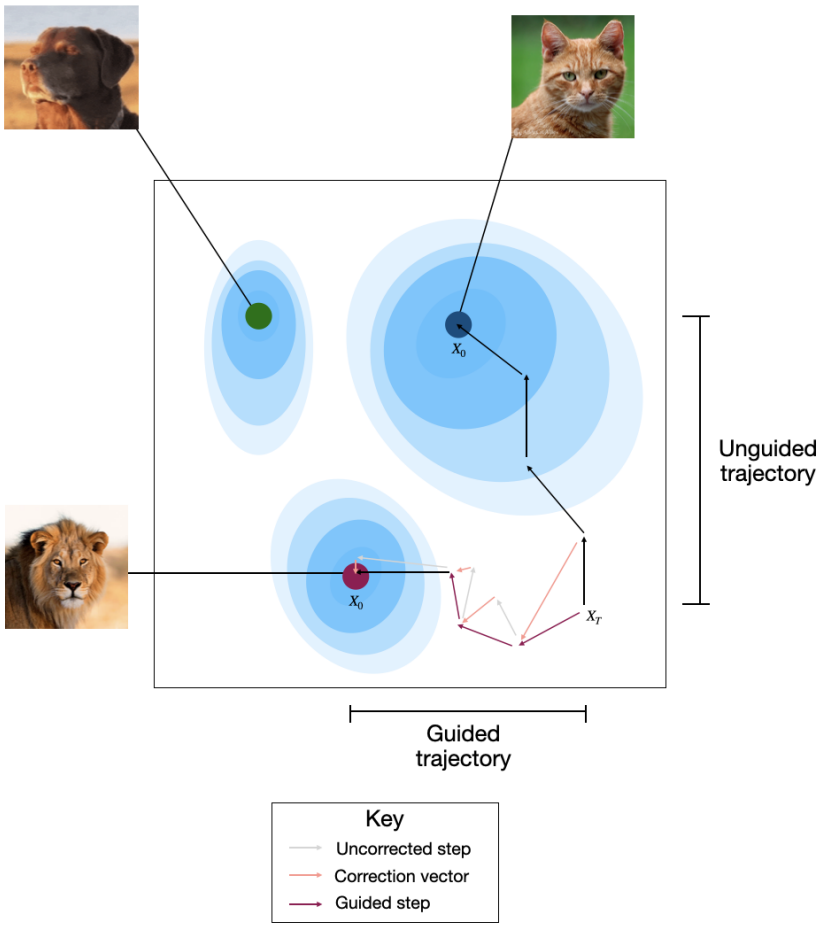


Figure 1.9: Illustration of guidance in generative models. The unguided trajectory (black) may drift toward undesired modes of the distribution. By introducing correction vectors (pink) at each step, the model follows a guided trajectory (purple) that steers sampling toward the desired target distribution (e.g., the lion).

have been developed. In the following, I describe classifier guidance (Dhariwal et al. 2021) and classifier-free guidance (Ho et al. 2022a), the former of which is employed in Chapter 5 to enable faster and more accurate inference of trimmed protein chains. Finally, I outline Feynman–Kac steering (Singhal et al. 2025), a more recent approach used in Boltz-1x (Wohlwend et al. 2025) and also applied in Chapter 5 to reduce side-chain clashes.

Classifier-based guidance relies on using a pre-trained classifier model that can guide a trained generative model, diffusion or flow matching, which predicts the score $\nabla_{x_t} \log p(x_t)$. By considering conditional generation as the score $\nabla_{x_t} \log p(x_t \mid y)$, the decomposition of the logarithmic terms can be written as:

$$\nabla_{x_t} \log p(x_t \mid y) = \nabla_{x_t} \log p(x_t) + \nabla_{x_t} \log p(y \mid x_t), \tag{1.3.7.12}$$

and so adding the gradient of the classifier term ($\nabla_{x_t} \log p(y \mid x_t)$) to the predicted score ($\nabla_{x_t} \log p(x_t)$) is sufficient to guide the output. In practice, though, the weighting of this guidance has to be identified (Dhariwal et al. 2021), and the noisy intermediate states are likely to be out-of-distribution relative to the clean target distribution that the classifier has been trained on (Ho et al. 2022a). Also, this type of guidance does not strictly require a classifier and instead can be achieved using any differentiable function, such as an energy function (Bansal et al. 2023).

Classifier-free guidance, in contrast, avoids the need for an external classifier to provide gradients during sampling (Ho et al. 2022a). Instead, the generative model itself is trained to handle both conditional and unconditional objectives by randomly omitting the condition during training. At inference time, conditional generation can then be expressed as a weighted combination of the conditional and unconditional predictions, with the weighting factor controlling the strength of guidance. This approach removes the dependence on a separately trained classifier and sidesteps the issue of noisy intermediate states being out-of-distribution for the classifier. However, the conditions available at inference must already have been specified during training; introducing a new condition later would require retraining the model. As such, classifier-free guidance offers more stability and simplicity in practice, but trades this off against reduced flexibility in accommodating new conditions.

Both types of guidance rely on the conditional being differentiable, whether that be by training the model or by using a differentiable function separately. However, if a desired condition is complex and not directly differentiable, Feynman–Kac steering can be utilised (Singhal et al. 2025). This methodology considers independent trajectories from a start point, scores them using functions that measure how well

the condition is being matched and then resamples trajectories using these scores to tilt distributions. This is advantageous when conditioning the model towards generating from rare yet desirable regions of the learnt data distribution.

Having established the fundamentals of ML, I next examine the impact of such methods on small-molecule drug discovery and highlight the central role of data in determining their success or failure.

1.4 The importance of data for small molecules machine learning

1.4.1 The impact of model architectural changes

There are great hopes that the ML and AI techniques described above could reduce the cost and increase the speed of drug development (Blanco-Gonzalez et al. 2023). ML has had significant success in the field of computer vision, (Ramesh et al. 2021; Croitoru et al. 2023); natural language processing, (Bubeck et al. 2023; Gozalo-Brizuela et al. 2023), and protein structure prediction (Jumper et al. 2021; Bertoline et al. 2023). In all these advances, data is abundant and sophisticated algorithms have been able to maximise what can be learned from it. The hope of the pharmaceutical industry and academia is that highly accurate ML models can be used to replace costly and slow experiments by producing reliable predictions or generating sensible hypotheses throughout the drug discovery pipeline (Lipinski et al. 2019). ML is a relatively low-cost methodology, requiring only computing resources and small numbers of specialist computational and data experts, as opposed to the equipment and facilities required for experimental screening and testing. Utilising ML methods to search for potential drugs in the enormity of chemical space (Reymond 2015) in silico is far more tractable than doing so in the lab, in vitro or in vivo. However, this potential tractability must be matched with sufficiently high accuracy. There are many areas of small molecule drug discovery which are already using ML including de novo molecular design, the design of novel molecules with desired properties (Meyers et al. 2021); retrosynthesis prediction, which aids in the planning of efficient synthesis routes (Jiang et al.

2022); docking, the prediction of 3D coordinates of ligand atoms bound to a target to inform molecular design (Sánchez-Cruz 2023); and property prediction, where crucial molecular properties are estimated by models (Mitchell, John BO 2014).

The development of ML methods in the field tends to follow a typical pattern where new architectures are proposed, trained with a popular training set, and tested on popular benchmarks and test sets (Su et al. 2018; Lowe 2012; Wu et al. 2018; Mysinger et al. 2012; Francoeur et al. 2020). This framework often results in only incremental improvement against older methods that were also trained and tested in the same way. To demonstrate this trend, I took three popular benchmarks for small molecules ML methods and compared their accuracy against their publication date. These benchmarks were 1) CASF 2016, a set of 285 protein-ligand complexes used to benchmark the ability of machine learning-based scoring functions (MLBSFs) to predict binding affinity for any protein-ligand complex (Su et al. 2018); 2) the USPTO-50k test set used to validate one-step retrosynthesis tools, that predict the reactants of compounds scraped from the US Patent Office (Lowe 2012), and 3) an HIV data set used to validate quantitative structure-activity relationship (QSAR) methods by classifying activity against the HIV protease from the MoleculeNet set of benchmarks (Wu et al. 2018). The results are shown in Figure 1.10 and show the same trend: there is little or no improvement using these benchmarks for both generalisable and specific small molecule ML methods. The exception to this is USPTO-50k, where a strong relationship can be observed, showing improvement over time ($\rho = 0.87$). However, this benchmark inadequately captures the true overall accuracy of retrosynthesis tools for full reaction path finding; the reasoning behind this is discussed further in Section 1.4.4. Therefore, this positive trend does not neccessarily reflect a real improvement in retrosynthesis tools.

While GNNs have achieved notable successes in other domains such as materials discovery (Merchant et al. 2023) and even delivered breakthroughs in drug discovery, including the identification of novel antibiotics (Stokes et al. 2020; Wong et al. 2023), their overall impact across small-molecule drug discovery has been more limited, as illustrated in Figure 1.10 and supported by other studies (Jiang et al. 2021a; Korolev

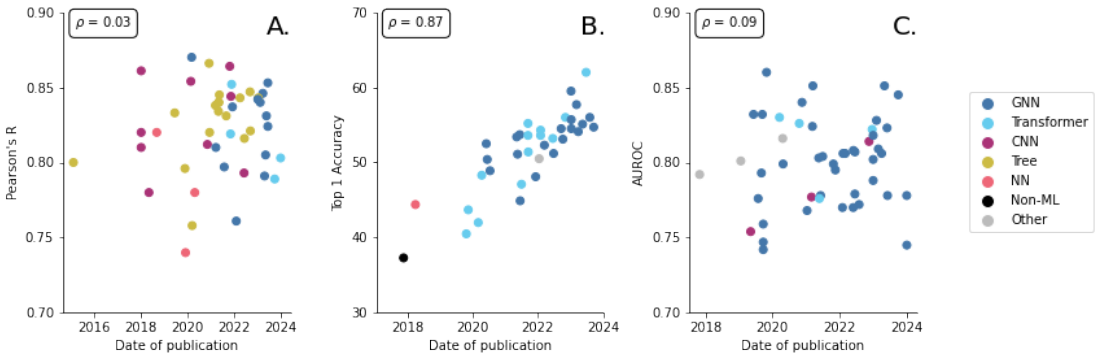


Figure 1.10: Benchmark performance with respect to publication date. The three benchmarks are A. CASF 2016 (Su et al. 2018), measured by Pearson's R, the linear correlation between predicting and true binding affinity; B. USPTO-50k, measured by Top-1 Accuracy, indicating the percentage of times the highest-confidence reaction was correct (Lowe 2012) and C. HIV MoleculeNet, measured by AUROC (Area Under the Receiver Operating Characteristic), a performance measurement for classification for binding or not binding at various thresholds.(Wu et al. 2018). Papers are coloured by ML architecture employed for the method, and Pearson's R (ρ) between the date of publication and the benchmark metric is given in the upper left box for each graph.

et al. 2020). A further illustration of machine learning architectures not working "out of the box" in the small molecules space is the application of diffusion to docking and conditioned de novo molecule generators (Corso et al. 2022; Igashov et al. 2024; Jing et al. 2022; Schneuing et al. 2024). These models have had mainstream success and are now used generally to generate artwork and images (Reed et al. 2023; Yildirim 2022; Azuaje et al. 2023). However, two studies have shown that despite increased accuracy compared to other methods, diffusion models for small molecules fail to generate physically plausible molecules or sensible interactions with proteins (Buttenschoen et al. 2024; Harris et al. 2023). These limitations can be partially alleviated through guidance or post-generation energy minimisation. However, such corrective steps increase computational cost, may distort the learned distribution, and ultimately fail to address the root issue that the models do not inherently capture simple physics (Wohlwend et al. 2025; Buttenschoen et al. 2024). All of this suggests that new ML architectural developments are unlikely to cause a step change in out-of-the-box applications. Instead, it will be a combination of ML methodology development, architectural and engineering decisions and the quality and quantity of data they are trained and validated on.

1.4.2 The data quantity problem

ML methods tend to require significant amounts of data to successfully model the underlying trends in the data (Sun et al. 2017). The success of large language models and image generation algorithms can be attributed to the abundance of text and image data that can be scraped from the entire internet. For example, the curation of training data for DALL-E 3 (Betker et al. 2023) or GPT-4 (Bubeck et al. 2023), where data is created passively by everyone using the internet every day. ML models trained on biological and chemical data exist in a very different space, as experimentally generating data is expensive, labour-intensive and difficult to automate, limiting our ability to scale up the amount of training data. This limits the available amount of training data to the order of tens or hundreds of thousands (Liu et al. 2014; Wu et al. 2018; Burley et al. 2017; Zdrazil et al. 2024), far smaller than the hundreds of millions for image generators (Ramesh et al. 2022) and trillions used to train large language models (Touvron et al. 2023).

AlphaFold 2 and other protein structure prediction tools (Jumper et al. 2021; Baek et al. 2021; Lin et al. 2023) leveraged the far larger corpus of publicly available protein sequence data (Mitchell et al. 2020; Suzek et al. 2007) to help overcome the relative paucity of experimentally determined protein structures (of the order of hundreds of thousands). The training data for small molecule methods in the public domain is often limited to tens to hundreds of thousands. A large amount of the data produced in this field is kept private by companies to protect their intellectual property. This siloing of data limits the field to relying primarily on whatever public data exists, often produced by academics or the limited data released by pharmaceutical companies for patents, such as KIBA (117,657 drug-target interaction pairs) (Tang et al. 2014), USPTO Yields (853,638 reaction yields) (Lowe 2012), Tox21 (8575 compounds) (Huang et al. 2016) and PDBBind (19,443 protein-ligand complexes) (Liu et al. 2014). Addressing this data scarcity is crucial for advancing ML in small molecule research.

One example, negative data, is very rarely publicly available, whether that be a lack of binding to a target by a compound (Réau et al. 2018) or a failed experiment

for organic synthesis (Strieth-Kalthoff et al. 2022). This deficiency is often attributed to publication bias, writing only about positive results and often not releasing or properly curating negative ones (Strieth-Kalthoff et al. 2022; Mlinarić et al. 2017). However, for training ML models, this creates a significant covariate shift in the proportion of positive to negative data points and so a classification imbalance. One example of this imbalance hampering accuracy is in yield prediction models, which are crucial for predicting the viability of experimental conditions and reagents for reactions in the synthesis of small molecule drugs (Strieth-Kalthoff et al. 2022). Pharmaceutical companies often have the opposite problem and have to downweight negative data or oversample positive data to improve training convergence time, such as when training DNA-encoded libraries (McCloskey et al. 2020), but typically this data is not made public. Proposed solutions such as stricter publishing guidelines to promote transparency, reproducibility, and accessibility of chemical data (Maloney et al. 2023; McEwen et al. 2023; Steinbeck et al. 2020), synthetic negatives (Segler et al. 2018), and crowd-sourced data such as the Dark Reactions Database (Ball 2020) have made limited headway in addressing this issue or can only be applied to prospective publications. Critically, though, addressing this deficiency will not be sufficient to improve ML methods for small molecule development, as the quality of the data is as important as how much is available.

1.4.3 Accounting for data quality

ML models are designed to identify and learn trends and patterns from their training data. However, ML models might learn spurious trends in the training data that may not be representative of real-world patterns, called data biases. Learning these data biases instead of relevant trends can result in reduced accuracy and utility of methods (Wang et al. 2023; Van Giffen et al. 2022; Leavy 2018; Lee 2018). This is especially relevant for small molecule research where data has often been curated from a variety of sources to mitigate the data quantity problems discussed above (Subramanian et al. 2016; Martins et al. 2012; Delaney 2004). Data points from each source are typically chosen by researchers, often exploring

limited areas of chemical space instead of random, unbiased sampling (Xie et al. 2022). For instance, in medicinal chemistry, there may be anthropogenic biases stemming from scientists' preferences for synthetic pathways and reactions. These preferences are influenced more by familiarity than by factors such as effectiveness, reduction of cost or ease (Thakkar et al. 2023).

Another type of data bias is derived from the dataset itself, rather than humans, called inductive bias (Cleves et al. 2008). The most prominent example of this is ML-based virtual screening tools, used to classify the activity of ligands to proteins. Many of these have been shown to rely on biases within the data, such as only relying on ligand-based features, rather than learning the patterns of protein-ligand interactions (Chen et al. 2019; Sieg et al. 2019; Jacobsson et al. 2006; Chaput et al. 2016). To create datasets that do not reward models memorising the ligands they are trained on, asymmetric validation embedding was proposed to generate tougher training-validation splits to penalise learning biases (Wallach et al. 2018; Tran-Nguyen et al. 2020). These more rigorous data splits have shown that current methods are not as accurate as originally presented (Jiang et al. 2021b; Shen et al. 2023), reinforcing the potential of methods that learn more than dataset biases. Adopting strategies or techniques to account for these biases could make ML models more robust to data distribution shifts (Klarner et al. 2023), with the identification of these biases an important necessary first step.

Another major limitation of small molecule data is the high amount of noise present in the datasets. Typically, this can be ascribed to the combining of data from various sources due to the restrictions on the amount of data that can be produced at once. A common example is the amalgamation of IC_{50} activity data from different assays to train a single QSAR model (Kramer et al. 2008; Kausar et al. 2018; Simeon et al. 2019) despite the different assay conditions making them not exactly comparable (Kalliokoski et al. 2013). Previous analysis has shown that IC_{50} and K_i for the same target and compound from different assays are correlated with the square of Pearson's R of 0.53 and 0.81 for each respective measure, showing the high uncertainty in the ground truth labels (Kramer et al. 2012; Kalliokoski et al.

2013; Landrum et al. 2024). This uncertainty can lead to inflated accuracy metrics for ML models, such as the maximum accuracy of MLBSFs on CASF 2016 (expanded on below) being estimated to be Pearson's R of 0.76 (see Figure 1.10) (Hernández-Garrido et al. 2023). Unfortunately, uncertainty estimates are not often curated in datasets such as ChEMBL (Zdrazil et al. 2024), necessitating approximations based on the type of data or assay. Training models to recognise and account for this uncertainty (Speck-Planche et al. 2022), and ensuring that the noise in the data does not compromise model accuracy, could help deal with this data problem.

Another source of noise is false positives, where experimental methods produce an incorrect positive readout or a non-reproducible result. One well-known example of compounds that generate false positives is pan-assay interfering compounds (PAINs). These are a set of compounds that appear to interact with multiple targets non-specifically, but often interfere with the mechanism of the assay directly through either redox-cycling or aggregation. This results in the compounds falsely appearing to be active against a specific target (Baell et al. 2018). Methods exist to predict whether a molecule could be a PAIN based on substructure flags or matches, such as the Brenk (Brenk et al. 2008) and NIH (Jadhav et al. 2010) catalogues of substructures. However, datasets are often not checked for such matches, for example, the widely used HIV Activity set from MoleculeNet (Wu et al. 2018) (see Figure 1.10), where of the 404 true actives, 70% had a match with these catalogues (Walters 2023). Though these matches are not proof of false positives, instead an indicator, nor is a lack of matches evidence of the opposite. Another example is for commonly used meta-learning benchmarks, where just picking out PAINs based on how many of the targets they hit was highly predictive (Klarner et al. 2022). These particular problems and sources of noise in small molecule data often require more specialised understandings of the origin of the data, such as experimental techniques and conditions, which are not immediately apparent to researchers with a more theoretical background. These examples highlight how subtle sources of noise and artefacts can pervade widely used datasets. Without careful curation and an awareness of their experimental origins, models risk learning

spurious correlations rather than generalisable patterns. Addressing these challenges is therefore essential for building reliable ML approaches in small-molecule discovery. However, to understand the impact of model engineering decisions, data quantity and quality, the most important part of small molecule ML model development, and arguably the one that receives the least attention, is validation.

1.4.4 Validating the methods

Validation of small molecule ML methods typically involves using a test set taken from the same source as the training set or testing on established benchmarks. Improvement, often incremental, is then used to justify the application of the model in prospective drug discovery (see Figure 1.10).

However, this approach often does not demonstrate whether the method is significantly more useful in drug discovery than existing methods, and in turn prevents the field from truly understanding what technological advances will help push method development forward. One underutilised method to help counter this is baselining and ablation testing of methods. By comparing to rigorous baseline methods, the value of adding model features and increasing model complexity can be justified. For example, work exploring potency prediction models using assay datasets from ChEMBL (Zdrazil et al. 2024) showed that a simple k-nearest neighbours baseline, which assigns targets based on the most similar ligands in the training set, outperformed graph convolutional networks and other ML models (Janela et al. 2022). This highlighted that many models were not learning trends beyond ligand similarity.

Often, methods in this field are evaluated on the types of data they are trained on, but would be used in a drug discovery setting for different kinds of data, a difference that is often not accounted for. An example of this limited evaluation is in retrosynthesis, where typically single-step methods are benchmarked on their accuracy on single-step reaction prediction, such as in the USPTO-50k dataset (see Figure 1.10). However, these methods are usually used with a path-finding algorithm, such as Monte Carlo tree search (Browne et al. 2012), to find the

total reaction pathway for a compound. A disconnect between the accuracy of these single-step methods on the commonly used benchmarks and their ability to successfully find full reaction pathways when combined with a search algorithm has been identified in a recent study (Torren-Peraire et al. 2024). Furthermore, these benchmarks, like UPSTO-50k, were found to be inappropriately small for evaluating models that could be trained on much larger and diverse sets (Torren-Peraire et al. 2024). These limitations highlight the need for more rigorous approaches to validation and benchmarking that more closely replicate how models will be used in practical drug discovery projects.

Overall, these examples demonstrate the general data problems facing ML applied to small molecules research. There is still promise as there are clear avenues for progress as speculated in this section; however, more research is needed to establish a step-change in how impactful the application of this technology can be. In this thesis, I focus specifically on the structural data for protein-ligand complexes and also their associated binding affinity labels. Next, I explain in greater detail the available datasets for training docking algorithms and scoring functions, and what methods have been developed on these datasets.

1.5 Computer-aided structure-based small molecule drug discovery

Several paradigms guide pre-clinical small-molecule drug discovery, including phenotypic screening (Prior et al. 2014) and ligand-based design (Ajjarapu et al. 2022). Among these, and the focus of this thesis, is structure-based drug discovery (SBDD) (Batool et al. 2019), in which 3D information about protein–ligand complexes is used to rationally design or discover molecules with improved therapeutic properties. Within SBDD, key methodologies include: docking (Morris et al. 2008), the prediction of a ligand's binding pose and its molecular interactions with a protein target; binding affinity prediction (Meli et al. 2022), the estimation of the strength of binding of any ligand binds to any protein; and de novo molecular design (Meyers et al. 2021), in which entirely new molecules are generated to bind

a specified protein with high affinity. The latter is not the focus of this thesis and so is not expanded upon. Recent advances in protein structure prediction, such as AlphaFold2 (Jumper et al. 2021) and protein–ligand co-folding models (Abramson et al. 2024; Wohlwend et al. 2025), have expanded the scope of SBDD by enabling reliable structural predictions even when experimental structures are unavailable. Training these methods requires datasets of protein–ligand complexes, whether experimentally determined, computationally predicted, or augmented with labels such as binding affinity. Here, I review the datasets commonly used for training and benchmarking these methods and highlight their limitations and deficits.

1.5.1 Datasets

Experimentally-determined structures

Protein Data Bank Structures of protein-ligand complexes that are published and publicly available are typically deposited in the PDB (Burley et al. 2017). Determination of structures can be done experimentally using techniques such as X-ray crystallography (Ilari et al. 2008), nuclear magnetic resonance (Wuethrich 1989) and cryo-electron microscopy (Fernandez-Leiro et al. 2016). However, all structures are deposited here, meaning other protein structures that interact with other compounds such as nucleic acids, glycolipids or other proteins. Therefore, filtering and data cleaning are required to obtain a dataset of just protein-ligand complexes.

PDBBind One of the most popular datasets for training machine learning models (Wang et al. 2005). It is manually curated from the PDB and originally was open to be used by any academics, but in 2022, it was taken behind a paywall. The developers updated annually until 2020, with the final publicly available version consisting of ∼20,600 protein-ligand complexes. Criteria for inclusion in this dataset include a published K_i, K_D or IC_{50} value that can be associated with the structure, and the structure has high enough quality. These quality control criteria and the manual curation are what limit the dataset size.

BindingMOAD This dataset was similar to PDBBind, but without the strict requirement for associated binding affinity (Hu et al. 2005). In its final update, it

consisted of ∼41,400 structures, of which 37% had an associated binding affinity label (Wagle et al. 2023). However, it also relies on manual curation or checking and will no longer be updated alongside PDB growth by the authors. Due to the larger dataset size compared to PDBBind, made possible by not requiring affinity labels for inclusion, this is a popular choice for docking algorithms as a training set (Corso et al. 2023).

PLINDER This dataset was proposed recently to address the critical demand for large protein-ligand structure datasets (Durairaj et al. 2024). The developers effectively curate any protein-ligand interaction where the ligand has an associated Biologically Interesting Molecule Reference Dictionary (BIRD) reference (Dutta et al. 2014), which includes metal ions and crystallographic agents. This less stringent filtering resulted in ∼613,000 examples of protein–ligand interactions (discounting artefacts and metal ions), corresponding to ∼158,000 unique protein–ligand pairs. However, the users can subsample the data or further clean to remove datapoints considered superfluous. AlphaFold 2 models of each holo structure were also generated (Jumper et al. 2021), and any apo conformations were linked to their respective holo structure. As well as training data, the accompanying publication provided increasingly challenging train-test splits (Durairaj et al. 2024). This dataset represents near-complete coverage of extractable protein-ligand interactions in the PDB. However, structural datasets have been developed that have gone beyond the experimental data of the PDB. I discuss popular examples below.

Predicted and augmented structures

Redocked2020 and CrossDocked2020 Initially, these two datasets were designed to train the pose classifiers and scoring functions GNINA (McNutt et al. 2021) (described below). By docking ∼20,600 protein-ligand complexes into their cognate structures (redocking) and docking into different holo structures (crossdocking) using Smina, a fork of AutoDock Vina (also described below), both examples of accurate poses and inaccurate poses could be trained on (Francoeur et al. 2020). To further augment the data and account for docking failure to produce a correct pose for a

given protein-ligand complex, the crystal structure pose was minimised with the UFF (Casewit et al. 1992) and Smina forcefields (Koes et al. 2013). Additionally, CNNs trained on the initial docked poses were used for pose generation, as scoring functions, through gradient descent, to generate adversarial poses that were inaccurately classified. This resulted in dataset sizes of 1.6m for Redocked2020 and 22.5m for Crossdocked2020 poses in the latest update (McNutt et al. 2025). As this methodology has produced augmented correct poses in altered conformations, this dataset has also been subsampled to correct poses within 1Å RMSD of the crystal pose for training de novo molecule generators (Schneuing et al. 2024). However, it is still limited to the initial ∼20,600 unique protein-ligand complexes docked.

BindingNet To go beyond the protein-ligand structures that have been structurally determined, the developers of BindingNet (Li et al. 2024; Zhu et al. 2025) leveraged the binding data from ChEMBL (Gaulton et al. 2012), where the outputs of assay results of different publications are curated and mapped to existing protein structures. To generate poses, they aligned only assayed ligands similar to the original ligand, using a Tanimoto similarity of 0.7 and optimised the pose. This resulted in a larger dataset of ∼70,000 protein-ligand complexes with a binding affinity label each, with greater enrichment for activity cliffs. However, due to the requirement for an existing similar protein-ligand complex, this dataset does not explore far beyond the protein and ligand space explored by existing structural datasets, although a later version did relax this requirement (Zhu et al. 2025).

1.5.2 Benchmarks and test sets

CASF 2016 CASF was a series of competitions held in 2007 (Cheng et al. 2009), 2013 (Li et al. 2014) and 2016 (Su et al. 2018) that tested the ability of scoring functions to predict binding affinity, rank binders and rank generated poses from docking software. The complexes used in the dataset were curated by taking the most representative clusters (90% sequence identity) of the PDBBind dataset and manually sampling five from each that covered the range of binding affinities in each cluster. The final version, CASF 2016, consisted of 285 protein-ligand complexes

and was docked to generate poses using MOE (Vilar et al. 2008), Gold (Verdonk et al. 2003) and Surflex (Spitzer et al. 2012). Its development was primarily focused towards physics-based functions but was adopted by MLBSF researchers (Meli et al. 2022); however, its deliberate high similarity to the training set provides an over-optimistic evaluation performance.

Astex Diverse Set A similar benchmark to the CASF 2016 set as it selected well-represented protein clusters in the PDB (Hartshorn et al. 2007). Ligands for each cluster were picked based on pharmaceutical or agrochemical interest and further manual curation. In total, it consists of 85 protein-ligand complexes, of which 75 have an associated binding affinity. It has been predominantly employed to assess the accuracy of docking software rather than binding affinity prediction (Buttenschoen et al. 2024).

2019 Holdout Developed to address the limitations of benchmarks like CASF 2016 and Astex Diverse Set that were not rigorous for benchmarking ML-based methods (Volkov et al. 2022). The authors proposed that a time split of PDBBind with all complexes after 2019 was a more appropriate test set, reflecting the nature of drug discovery campaigns where decisions are based on collected existing knowledge. This test set was utilised to assess MLBSFs and showed that they perform less accurately compared to on CASF 2016.

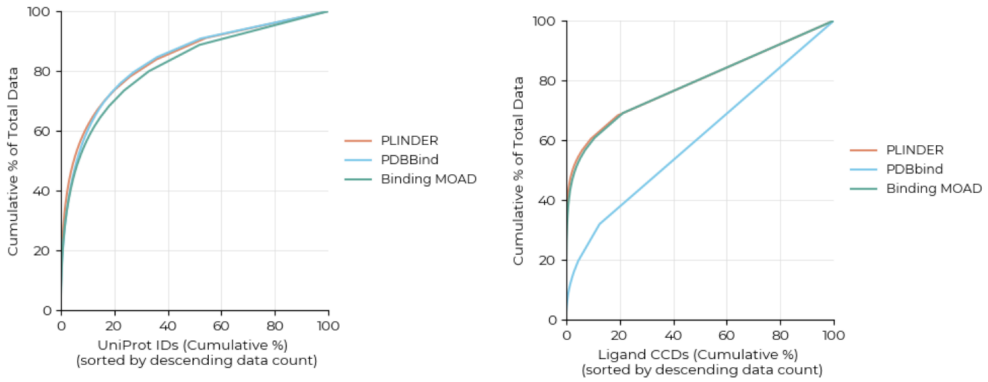
PoseBusters Benchmark An additional time-split-based benchmark developed for docking software (Buttenschoen et al. 2024). Like the 2019 Holdout, the authors used data only after a strict time cutoff of 1 January 2021 from the PDB, so there was no overlap with methods trained on PDBBind. Subsequent quality control and filtering, so that each protein-ligand complex had a unique UniProt ID, left the final dataset, which consisted of 308 complexes. To provide more rigorous analysis of performance on dissimilar data, the dataset was further split into subsets (0-30%, 30-95% and 95-100%) according to sequence similarity to data in PDBBind.

Runs N' Poses One of the more recent benchmarks developed, the authors again utilised a time-test cutoff of 30 September 2021, specifically as a common training date cutoff for the co-folding method (Škrinjar et al. 2025). By leveraging

the filtering and metadata provided in PLINDER, ∼2600 protein-ligand complexes were curated to form the Runs N' Poses set. Further clustering by pocket similarity, measured using normalised SuCOS scores (Leung et al. 2019), created subsets, as done for the PoseBusters benchmark.

1.5.3 Biases and diversity problems

The field relies heavily on the PDB as the primary source of structural data to train machine learning methods, with the implicit assumption that it provides an unbiased sample of all protein–ligand interactions. In reality, experimental structure determination is expensive and guided by practical considerations, meaning that the PDB reflects strong trends and biases. Certain proteins and ligands are heavily overrepresented because they are common drug targets or of particular interest to the biological community, as shown in Figure 1.11. As a result, the training data



Dataset	% UniProt IDs in 50% of total data
PLINDER	4.9
PDBBind	6.0
BindingMOAD	6.7

Dataset	% Ligand CCDs in 50% of total data
PLINDER	2.4
PDBBind	35.7
BindingMOAD	3.4

Figure 1.11: Comparing PLINDER, PDBBind, and BindingMOAD. The cumulative distributions of UniProt IDs (left) and ligand CCDs (right) illustrate redundancy across datasets. PLINDER was filtered to contain unique protein–ligand pairs with ions and common artefacts removed, whilst BindingMOAD and PDBBind were left as is. This highlights differences in dataset design and composition that affect downstream benchmarking and modelling.

is not broadly diverse but skewed towards specific proteins and ligands. Notably,

the datasets exhibit similar diversities across ligand and protein space despite the variation in size. The only exception is PDBBind for ligand diversity, which shows higher diversity, shown as a flatter curve, demonstrating that the data is more spread across the ligand space. This is likely due to the restrictions of binding affinity excluding common groups, such as FAD or NAD, that PLINDER and BindingMOAD do not. While this focus can be advantageous for drug discovery efforts centred on these targets, it limits generalisation to novel proteins or chemical spaces.

Biases also arise from the preferences of medicinal chemists and the trajectory of drug discovery campaigns. For example, binding affinity trends often reflect the presence of common substituents such as fluorines, methyl groups, or fragment-sized scaffolds, shown in Figure 1.12. Some of these biases are useful; small fragments

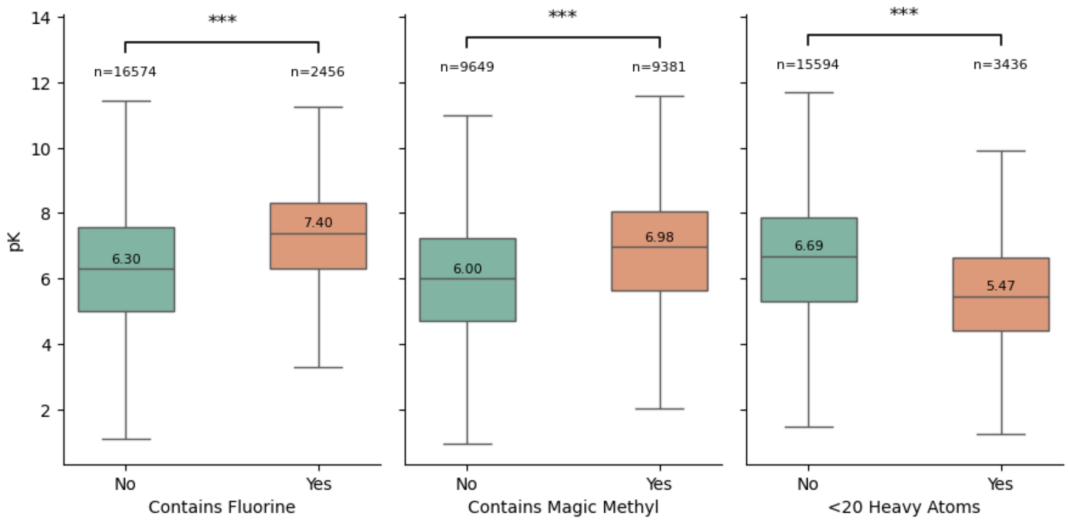


Figure 1.12: Distributions of binding affinity values (pK) in PDBBind, stratified by molecular features. Boxplots show the effect of (left) fluorine substitution, (middle) the presence of a "magic methyl" group, and (right) ligand size with fewer than 20 heavy atoms. Median values are displayed within the boxplots. Statistical significance (p=0.0001) is indicated by asterisks (***) measured by Mann-Whitney U test.

do generally bind weakly, so learning this prior aids prediction (Giordanetto et al. 2019). Methylation, the so-called "magic methyl" effect, is often pursued to increase hydrophobicity and binding affinity (Pinheiro et al. 2023). Yet structures are usually reported only when methylation improves affinity; unsuccessful modifications, which may create steric clashes and weaken binding, are under-represented. Similarly,

fluorination frequently appears correlated with binding affinity, but this is largely an artefact: fluorination is typically used to improve metabolic stability in lead optimisation, so its presence in solved structures reflects downstream selection rather than a causal effect on binding (Yerien et al. 2016).

These examples illustrate how dataset biases shape the signal available to models. While identifying every such bias is infeasible, acknowledging and accounting for them is essential if structure-based ML methods are to generalise beyond the narrow distributions present in the PDB. Next, I explore the different classes of methods that utilise these datasets for prediction for SBDD and how ML has been used to improve them.

1.5.4 Binding affinity prediction

Predicting binding affinity is a central goal in computational drug discovery, as knowing the ability of a compound to bind a target is crucial for determining therapeutic potential (Kairys et al. 2019). Binding affinity prediction provides a more general and transferable measure of interaction strength that can also be utilised for hit-to-lead and lead optimisation stages of the drug discovery pipeline (Jorgensen 2009). It can also be used in prioritisation of compounds in virtual screening campaigns, where the task is often to rank or classify molecules by their likelihood of binding (Kim et al. 2008). However, dedicated virtual screening methods may be more directly applicable (Zhou et al. 2024).

Experimentally, binding affinity is typically measured through biophysical constants such as the equilibrium dissociation constant (K_D), inhibition constant (K_i), or half-maximal inhibitory concentration (IC_{50}), which are reported in biochemical assays. Taking the negative logarithm of any of these constants gives the pK, with higher values representing higher binding affinities. However, these methodologies are not sufficiently high-throughput to evaluate the vast chemical space accessible in drug discovery, motivating the development of computational approaches to predict binding affinity (Meli et al. 2022). Ligand-only prediction of binding affinity for a given protein, such as QSAR models (Tropsha 2010), is

popular but individual models often cannot be used across different drug discovery campaigns. Structure-based methods offer the possibility of predicting binding affinity for any given ligand and protein. Here, I discuss physics-based and machine-learning methods for this (depicted in Figure 1.13).

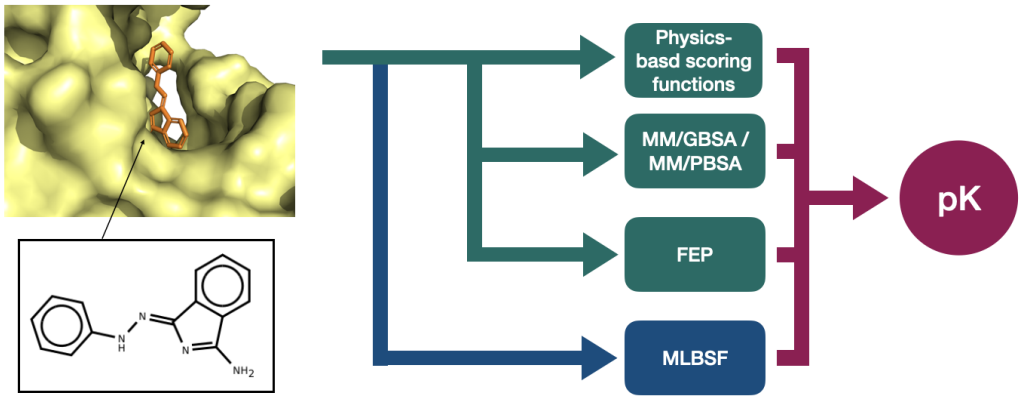


Figure 1.13: Overview of binding affinity estimation approaches. A docked protein–ligand complex can be evaluated using physics-based scoring functions, endpoint free energy methods (Molecular Mechanics using Generalised Born Surface Area (MM/GBSA), Molecular Mechanics using Poisson-Boltzmann Surface Area (MM/PBSA)), free energy perturbation (FEP), or MLBSF to predict binding affinity (pK).

Physics-based methods

Physics-based scoring functions Scoring functions are methods that predict a binding affinity given a single, static structure of any protein-ligand complex. Historically, these scoring functions could be classified as force-field-based, empirical or regression-based, and knowledge-based.

Force-field-based scoring functions relied on molecular mechanics forcefields to evaluate the protein-ligand interactions, such as DOCK (Kuntz et al. 1982), AutoDock (Goodsell et al. 1990), and GoldScore (Jones et al. 1997). These forcefields and their parameters were developed to reproduce ab initio quantum calculations or experimentally observed properties. By using these pre-existing and physically grounded functions, they can be readily updated with forcefield advances. However,

they are not explicitly aligned or trained to be accurate for binding affinity prediction and should instead be considered proxies instead of predictions.

Knowledge-based scoring functions use pairwise statistical potentials that give greater weight to interactions between two atoms found to be more frequent than them not interacting in a reference state, such as an ideal gas model (Muegge et al. 1999). Examples include DrugScore (Sotriffer et al. 2002), ITScore (Huang et al. 2006c) and the PMF scoring functions (Muegge et al. 1999). These interactions are curated from existing structures in large datasets in protein-ligand complexes.

Finally, empirical or regression-based scoring functions use regression analysis to fit pre-determined features to experimental binding affinity. They typically assume a linear relationship to features whose parameters are learnt. The features are often physics-inspired and small in number, such as ChemScore (Verdonk et al. 2003) and AutoDock Vina (Trott et al. 2010). This class can be considered a precursor to machine learning-based scoring functions, with simple statistical models being fit to data mirroring the training of complex ML architectures. All three classical approaches offer fast inference yet were limited in accuracy (Su et al. 2018) as the trade-off of speed usually resulted in oversimplifications or coarse approximations of the physics of protein-ligand binding. Molecular dynamics-based methodologies such as MM/GBSA, MM/PBSA and FEP offer such accuracy.

MM/GBSA and MM/PBSA MM/GBSA (Kollman et al. 2000) or MM/PBSA (Srinivasan et al. 1998) are methods that use ensembles of protein-ligand complexes under a classical forcefield for the prediction of binding affinity. They consider the free binding energy as a decomposition of energy terms that can be derived from the simulations, such as electrostatic and internal energies. The polar contribution to the energy can be calculated from the GB or PB method, with GB preferred for faster calculations. Non-polar contributions are inferred as a linear model of the solvent accessible surface area (SASA)). The accuracy of the method is variable, with length and number of independent simulations influential, but it offers a compromise of speed and accuracy for prediction (Genheden et al. 2010).

Free energy perturbation FEP methods combine molecular dynamics with statistical mechanics to estimate relative binding free energies, most commonly between congeneric ligands (Zwanzig 1954). The principle that allows FEP to be potentially highly accurate is that it uses a thermodynamic cycle of the ligand bound to the protein and unbound to get a difference in free energy. Calculating this free energy difference requires sampling many configurations or microstates and calculating their energies (E_i), from which a partition function (Z_{state}) can be calculated:

$$Z_{\text{state}} = \sum_{i}^{\text{state}} \exp\left(-\frac{E_i}{k_b T}\right) \tag{1.5.4.1}$$

where k_b is the Boltzmann constant and T is the temperature of the system. By evaluating the partition function for the bound ($Z_{complex}$) and unbound states ($Z_{solvent}$), the binding constant and, therefore, the binding energy (ΔG_{bind}) can be calculated.

$$\Delta G_{\text{bind}} = -RT \ln\left(\frac{Z_{\text{complex}}}{Z_{\text{solvent}}}\right) \tag{1.5.4.2}$$

R represents the universal gas constant. However, these absolute calculations can be inaccurate due to imprecise approximations of quantum mechanics and incomplete sampling of microstates. These limitations typically arise due to constraints on computational resources and time (Muegge et al. 2023). To overcome this, the relative difference in binding affinity can be measured using "alchemical perturbation" between the ligands, bypassing the need to consider the unbound states. This perturbation maps two closely related ligands, within 10 atoms in difference, and considers the free energies in the transformation to progressively transform ligand A to ligand B along a virtual coordinate (Muegge et al. 2023).

$$\Delta\Delta G_{A,B} = -RT \ln\left(\frac{\frac{Z^B_{\text{complex}}}{Z^A_{\text{complex}}}}{\frac{Z^B_{\text{solvent}}}{Z^A_{\text{solvent}}}}\right) \tag{1.5.4.3}$$

Machine learning-based scoring functions

MLBSFs are trained on experimentally determined or predicted protein–ligand complex structures paired with binding affinity labels, using either direct structural

representations or engineered featurisations (Meli et al. 2022). They leverage the curated datasets described above, such as PDBBind, to learn a generalisable understanding of binding affinity. By fitting a model to the data, the promise is that these methods can maintain the speed of physics-based scoring functions and reach the accuracy of molecular dynamics-based methods such as FEP (Valsson et al. 2025).

Initially, models were trained using basic architectures and designed features. RFScore was one of the foundational models to do so, by utilising the accuracy of random forest models and featurising the protein-ligand complex as counts of atom (Ballester et al. 2010). Specifically, the counts of whether an atom type or element within 12Åis within another type. This simple approach provided high accuracy and spurred the development of other methods. The authors developed this method further by exploring more complex descriptors (Ballester et al. 2014) and applying it to virtual screening (Wójcikowski et al. 2017). Another notable example includes NNScore (Durrant et al. 2011), which used more complex features, named BINANA features, that were input into a neural network. These are more complex than those utilised for RFScore, considering counts of atom types in different contexts, such as close contacts and electrostatics and also additionally using the terms of the AutoDock Vina scoring functions (Trott et al. 2010). There have been further interaction-based featurisations using fingerprints, such as PLEC (Wojcikowski et al. 2019) and complex topology modelling of the protein-ligand interactions (Liu et al. 2022). By adding 2D ligand descriptors to these features, accuracy was shown to be improved, but concerningly, they were highly predictive alone (Boyles et al. 2020).

Instead of using human-crafted features, models that learnt directly from featurisations of the protein-ligand complex structure were developed. Initially, these were built using CNN models that voxelised the structure into grids with atoms represented as occupancy features of each $1Å^3$ grid. Examples of this include KDeep (Jiménez et al. 2018) and Pafnucy (Stepniewska-Dziubinska et al. 2018), who differed mainly in how the voxels were featurised. GNINA can also be considered similar to these methods, but it was co-trained to classify poses, on the Redocked2020 and CrossDocked2020 datasets, and is trained using a psuedo-Huber

loss (McNutt et al. 2021). Voxelisation has limitations in that a fixed window of the protein-ligand complex is required as input, which can truncate larger ligands, and also, essential but small-scale atomic details can be coarse-grained. Hence, GNNs were applied to training scoring functions, and numerous methods have been proposed, each with unique properties. SIGN used polar coordinate-inspired graph attention to learn distance and angle information (Li et al. 2021), PointVS employed an EGNN and also pretrained on pose classification, with a focus on extracting proper attribution for fragment elaboration (Scantlebury et al. 2023). AEV-PLIG combined atomic-environment vectors and trained on augmented data to improve performance on ranking congeneric series (Valsson et al. 2025). To bake physics-based biases into the mode, PIGNet used GNNs to parametrise physics-based equations of interactions, which constrained the model to predict binding affinity whilst aligning with physical principles (Moon et al. 2022). The number of methods published that can be used or can be described as machine learning-based methods is substantial. The field receives high interest and effort due to the promise of making a substantial impact on small-molecule drug discovery. Research has shown frequently that these methods are poor at generalising to accurate predictions outside of their distribution, despite their apparent increase in accuracy over physics-based methods (Scantlebury et al. 2023; Volkov et al. 2022; Kramer et al. 2010). It is unclear whether the frequent development of new architectures addresses the deficits of MLBSFs (shown in Figure 1.10).

1.5.5 Docking

The prediction of the binding pose of a ligand is important for predicting binding affinity, as it determines the exact conformation and interactions of the ligand bound to the protein. Furthermore, these methods can be used to understand mechanisms of action of known binders and to build qualitative structure-activity relationships, guiding rational optimisation of lead compounds (Ferreira et al. 2015).

Physics-based methods

The first physics-based docking methods utilised their accompanying physics-based scoring functions to search and optimise across all possible poses of a ligand binding to the protein. The methods that relied on the best predicted binding energy would pick out the most accurate pose from multiple generated hypotheses. One of the first methods, DOCK, searched the possible conformational space by fitting spheres to the designated pocket and then matching ligand atoms to the centre of these spheres (Kuntz et al. 1982). The method did not account for ligand or receptor flexibility, and so was initially a rigid-body docking method. To improve performance, the search methods were improved instead using simulated annealing across the entire pocket and expanded to account for ligand flexibility as well, such as for AutoDock (Goodsell et al. 1990). Iterations and improvements were driven by more efficient search methods, such as the genetic algorithm, leveraged in GOLD (Verdonk et al. 2003) or gradient-based optimisations, utilised as AutoDock Vina (Trott et al. 2010).

To expand the capabilities of these methods to dock into inaccurate protein conformations, flexible side-chain docking was developed in the AutoDock suite and was adopted by other methods (Morris et al. 2009). These methods for modelling protein flexibility incur increasing computational costs with added side-chain flexibility, limiting their utility beyond modest levels of flexibility (Zhao et al. 2008). Constrained docking also enhanced the accuracy of these methods by enabling the specification of user-input constraints in the docking process. These could have been identified from previous structural experiments or from developed structure-activity relationships (Wang et al. 2019). Examples of constraints include specification of hydrogen bond formation between protein and ligand atoms and covalent bond formation (Bianco et al. 2016). All these different methods have been invaluable parts of the SBDD toolkit, yet still have room for improvement. One clear deficit is that these methods rely on the specification of a pocket or search space, as larger search spaces result in more false positive poses being identified or proposed and the method getting stuck in local minima (Ferreira et al. 2015).

Furthermore, the functions used to rank the proposed poses were not well calibrated, which obfuscated triaging of successful docking (Chang et al. 2010).

ML-enhanced docking methods

The introduction of ML to structure-based drug discovery also had a significant impact on docking, where, at first, it was used to enhance or augment existing docking methodologies. As described above, physics-based scoring functions are developed to approximate binding affinity and assume that inaccurate or poorer quality poses would have lower predicted binding affinity. However, often these functions were a sum of energy terms that correlated with the number of atoms, hydrogen bonds or rotatable bonds. To overcome this, ML methods were trained to rank or classify these generated poses and so improve the accuracy of these methods and, in some cases, provide well-calibrated predictions for the accuracy of a pose (Francoeur et al. 2020). Examples of this include GNINA (McNutt et al. 2021), which was trained for pose classification, as well as binding affinity, using the Redocked2020 and Crossdocked2020 datasets described above to enhance the docking accuracy of Smina (Koes et al. 2013), a fork of AutoDock Vina. Another example is DeepDock (Liao, Zhirui and You, Ronghui and Huang, Xiaodi and Yao, Xiaojun and Huang, Tao and Zhu, Shanfeng 2019) and its successor RTMScore (Shen et al. 2022), which learnt an interaction potential through a mixture density network for each ligand-protein interaction. By summing these predicted potential values, proxies for energy could be used that demonstrated strong performance in docking ranking and screening on the CASF 2016 benchmark. Optimising the pose using this potential was able to be used as a docking software, but it still relied on the traditional search process employed by physics-based docking software. Due to the need for traditional docking software to have a pocket or search space defined a priori for accuracy, ML methods have been developed that can identify these pockets, such as P2Rank (Kriválk et al. 2018).

ML-only docking methods

ML methods have been applied not just to augment the accuracy of physics-based methods but to replace them entirely. By utilising large unlabelled datasets, UniMol was able to learn from conformers of diverse molecules and could be finetuned to both rank poses and to generate the position of poses directly (Zhou et al. 2023). The author's evaluation found it was able to outperform traditional methods in producing poses below 2Å root mean squared deviation (RMSD), a standard accuracy threshold for docking. However, it is unclear whether its performance is improved by pocket definition in the preprocessing steps and its subsequent leakage (Buttenschoen et al. 2024). TANKBind leveraged trigonometric constraints and P2Rank to isolate specific pockets to dock into to enable blind docking to proteins with side-chain flexibility and prediction of binding affinities (Lu et al. 2022b). The model learns through GNN distance maps for the protein and ligand to get predicted ligand-protein distance maps, from which the coordinates of the pose are calculated using a numerical approach.

A novel approach to docking was to learn directly the ability to produce a pose given an entire protein, with no restrictions on the search space (depicted in Figure 1.14). This bypasses the need for prediction with pocket detection methods and promises the discovery of unexpected or cryptic binding sites that conventional pocket-specific methods could overlook. The first developed method, EquiBind, predicts a transformation for a ligand conformer that can be used to produce a rough estimate of a pose, which can then be minimised using physics-based functions (Stärk et al. 2022). However, its regression-based learning was found to be problematic if multiple pockets existed, justifying the development of DiffDock to learn a distribution of poses instead (Corso et al. 2022). This docking method was the first to use diffusion to generate possible poses by learning a mapping from a Gaussian to a data distribution of "all correct poses". One of the key innovations was the application of torsional diffusion and diffusion for translation and rotations to iteratively update the coordinates of the ligand pose whilst holding to chemical and physical inductive biases such as bond lengths and bond angles

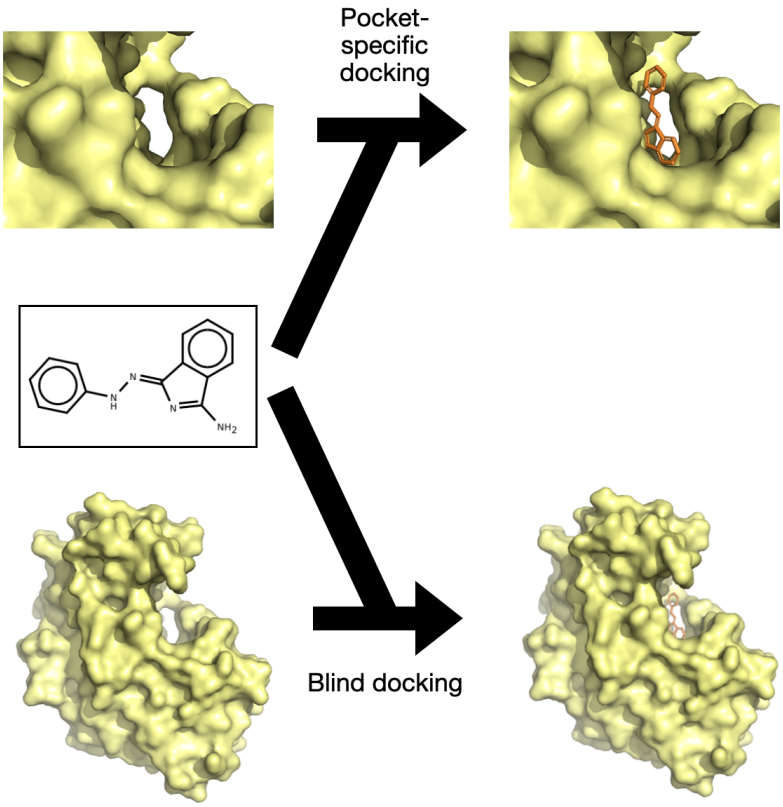


Figure 1.14: Comparison of docking approaches. In pocket-specific docking, the ligand is placed directly into a predefined binding site, whereas in blind docking, the ligand is positioned without prior knowledge of the binding pocket, requiring the model to identify and fit into the correct site.

remaining unchanged. Further method developments included SurfDock (Cao et al. 2025), which leveraged MaSIF-derived (Gainza et al. 2020) surface descriptors to improve pocket docking accuracy, and DiffDockPocket (Plainer et al. 2023), which applied diffusion-based approaches to pocket-specific docking and flexible side-chain prediction. Flow matching was also utilised to increase the speed of training and inference, but without substantial improvements in accuracy relative to diffusion (Morehead et al. 2025). Many of these methods still relied on generating multiple poses and ranking or scoring them, so often, independent pose scores, either based on predicting RMSD (DiffDock) or potentials like DeepDock.

Despite the impressive accuracy gains, issues have been identified with the plausibility of poses predicted and the generalisability of the methods (Buttenschoen et al. 2024). To address these concerns, minimisation and loss functions have

been proposed to both fix and improve the quality of the physical plausibility of poses (Wohlwend et al. 2025). However, one limitation of docking methods is the reduction in accuracy when using predicted or alternate protein conformation structures, as these methods are trained on the cognate structures. By treating the protein rigidly or with limited side chain flexibility, these methods cannot capture significant conformational shifts induced by binding or backbone. Co-folding methods have been proposed to predict both protein–protein complexes and protein–ligand interactions, explicitly accounting for the flexibility induced upon binding. To understand their development, it is useful first to consider their predecessors: protein structure prediction methods.

1.5.6 Protein structure prediction

AlphaFold 2 was a key breakthrough in the accurate prediction of protein structures (Jumper et al. 2021) that has enabled 214 million protein sequences to be structurally annotated (Varadi et al. 2024). Proteins are organised hierarchically: the primary structure is the amino acid sequence, the secondary structure consists of local motifs such as α-helices and β-sheets, and the tertiary structure corresponds to the full 3D fold of a single chain. Beyond this, the quaternary structure captures how multiple protein chains, and often non-protein components, assemble into functional complexes. Predicting tertiary and quaternary structure directly from sequence has long been the central challenge of protein structure prediction.

The methodology behind this success was a complex implementation that relied on an accumulation of different advances. Before AlphaFold 2, protein structure prediction methods leveraged co-evolutionary features extracted from multiple sequence alignments (MSAs). By identifying sequences that co-evolved, by identifying paired mutation patterns in different sequences, their proximity in 3D can be determined, effectively giving a priori knowledge of how the protein sequence folds to the prediction method (Hopf et al. 2014). Alphafold 2 directly took the MSA as an input and outputted directly the structure; this end-to-end approach demonstrated that by bypassing the human intuition used for featurisation, substantially greater

accuracy could be obtained. To do this, a large part of the model's parameters are part of the Evoformer block that learns sequence relationships both within the input sequence (row attention) and across the same position in different sequences (column attention). This model section iteratively updates both the MSA representation and pair representation across the sequence, leading to deeper relationships being learnt from just the sequences, before any structure prediction occurs.

AlphaFold 2 used several inductive biases baked into the model in order to predict the structure from the pair and single sequence representations. The first is that the residues of each part of the sequence are considered to be a fixed frame instead of the separate bonds and atoms they consist of. By doing this, the model has to predict a translation and orientation of each frame to predict a structure, reducing the likelihood of unphysical structures being produced, although minimisation with the AMBER99SB forcefield is still required (Showalter et al. 2007). To ensure that this prediction of local orientations and positions is invariant to global orientation and positions, the developers introduced invariant point attention (IPA). This uses attention to both the sequence embedding and individual residues and attention between residues, based on squared distances between their respective key points in 3D space, with these updated features used to predict frame updates. The utilisation of templates, predicting confidence in structures and using this to train on additional high-confidence prediction (self-distillation) were also significant drivers of the accuracy of this method. Other methods were developed that improved or optimised performance for specific protein classes, such as by using language model embeddings instead of the MSA (Lin et al. 2023) or training exclusively for antibody structures and dropping the need for MSA entirely (Abanades et al. 2023). AlphaFold Multimer extended AlphaFold 2's single-chain predictions to be applied to protein-protein complexes (Evans et al. 2021).

The accurate prediction of protein structures, using methods such as Alphafold 2, promised the ability to dock small molecules to a much wider range of structures. However, as discussed above, physics-based and ML docking methods were found to have lower accuracy on such methods, and subsequently, binding affinity prediction

and virtual screening performance were also harmed (Scardino et al. 2023; Wong et al. 2022). One cause for this was found to be due to poor side-chain orientation and pocket loop prediction. Furthermore, AlphaFold 2 cannot predict co-factors, sugar groups and metal ions, which can be important for ligand binding. This was the justification for the prediction of the ligand and protein complex together, to accurately predict any biological complex (shown in Figure 1.15). Next, I describe

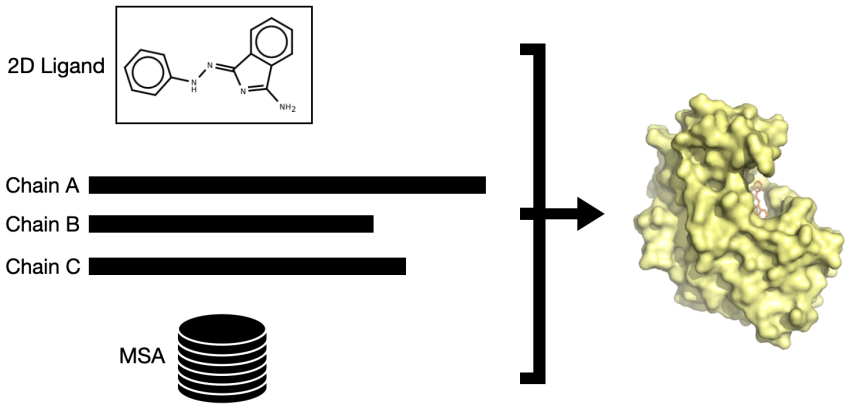


Figure 1.15: General schematic of cofolding. Protein sequences, MSAs, and ligand information are provided as inputs, and the model predicts the folded protein-ligand complex.

the development of the cofolding method, AlphaFold 3 (Abramson et al. 2024), and the subsequent open-source methods that aimed to replicate their performance.

1.5.7 Cofolding

AlphaFold-3 was not the first method to predict the structure of protein and ligand concurrently, with DragonFold (Scheen et al. 2025) and RFDiffusionAA (Krishna et al. 2024) being the first methods to do so. However, it was the first cofolding method to demonstrate ligand pose prediction accuracy exceeding existing docking software despite having to predict the protein structure accurately as well as the ligand pose (Abramson et al. 2024). The architecture (depicted in Figure 1.16) inherited similarities to AlphaFold 2, such as being an end-to-end method that took in the sequence and MSA of the input protein sequences, but to manage

the prediction of non-protein groups, significant changes were introduced to the architecture. Proteins were still preprocessed as residue tokens, but DNA and RNA

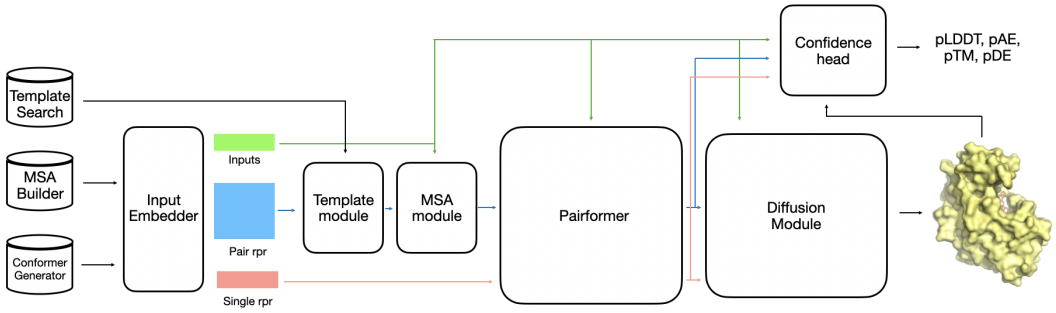


Figure 1.16: Schematic overview of AlphaFold3's architecture. Protein sequences and ligand conformers are embedded with MSAs and template information, processed by the PairDormer and diffusion modules, and evaluated with confidence metrics (pLDDT, pAE, pTM, pDE) to generate a protein–ligand complex structure. Adapted from (Abramson et al. 2024)

were split into their nucleotides and ligands, their atoms. To enable the tokens of the protein sequence to be attended to along with the tokens of the non-protein components, the PairFormer was developed to replace the EvoFormer to update the pair and single representations. MSA processing was substantially de-emphasised and kept separate (MSA Module), with only the first row carried through to the PairFormer. The PairFormer still leveraged the same triangular multiplicative and triangular attention update, initially introduced in the EvoFormer, to encourage residue-residue distances and angles to be kept geometrically consistent. One of the major differences was the removal of the use of inductive biases, such as IPA and prediction on frames. Instead, positions are directly predicted using atomic diffusion (Diffusion Module) with no equivariant or invariant biases baked into the method. As atomic coordinates are what are directly predicted, the model uses a MSE loss instead of the frame aligned point error (FAPE) loss used in AlphaFold 2. AlphaFold 3 was able to learn equivariance through randomly rotating the noised input coordinates throughout training, showing that models are able to understand these biases directly from the data without constraining the model. Additionally,

several confidence metrics were trained (pLDDT, pDE, pAE and pTM) to provide estimates of the accuracy of the output structures.

The accompanying publication demonstrated strong accuracy across many biological complex structure prediction tasks, such as protein-antibody complex prediction and protein-DNA prediction. However, a major limitation of the method was its restrictive licensing and the lack of available code, with only a web server provided with limited functionality. This prompted several efforts to replicate the methodology whilst keeping it open-source, Boltz (Wohlwend et al. 2025), Chai-1 (Chai-Discovery-Team et al. 2024) and Protenix (ByteDance et al. 2025). These implementations have not been able to replicate the performance of AlphaFold 3 perfectly, but have been taken up by the scientific community (Škrinjar et al. 2025). This could be due to subtle differences in implementation, such as Boltz using all intermediate denoised structures in the confidence head (Wohlwend et al. 2025). In Chapters 3 and 5, I utilise Boltz-1x for the prediction of "apo" structures of complexes and for the development of new methods for the prediction and hallucination of ligand-binding protein pockets. This enhanced the physical plausibility of its generations, compared to Boltz-1, using simple potentials with diffusion guidance and steering. A newer Boltz model (Boltz-2) has been trained to predict binding affinity for protein-small molecules, additionally (Passaro et al. 2025). It was trained on a large subset of assay data available in ChEMBL instead of structures themselves and was found to be close to FEP performance. Yet it is unclear currently if Boltz-2 has addressed the limitations of MLBSFs described above, such as poor generalisation and tendencies to memorise. The performance of these co-folding methods has also been shown to decline with data outside of their training distribution (Škrinjar et al. 2025), reflecting trends in ML for docking and scoring functions described above. Generalisation to novel ligands and proteins remains a problem for all these classes of methods.

1.6 Thesis outline

In this chapter, I have outlined the small molecule drug discovery pipeline and the principles of ML, which promise to improve its efficiency. I then explored how the quality and coverage of data and validation are as important as the algorithm choice for driving success in applications to small-molecule drug discovery. Finally, I explore structure-based drug discovery and how ML has been applied to improve accuracy, highlighting the datasets used and their deficits.

In Chapter 2, I interrogate MLBSF performance to understand whether the models are learning more than the underlying biases present in the data that trained them. I found that baselines, designed to learn only from the dataset biases, matched the performance of diverse methods, indicating that MLBSFs are unlikely to be learning the underlying physics of binding. Further benchmarks also highlight this tendency to learn dataset biases.

In Chapter 3, I explore the robustness of pose-classifiers for docking software to noisy poses and discover that slight noise to the ligand pose can impair confidence in poses. I find that training on this noise rescues accuracy. However, this does not result in improved accuracy in scoring poses from other docking software, which is also out-of-distribution.

In Chapter 4, I evaluate the capabilities of ligand pocket generation methods to produce synthetic protein-ligand complexes by predicting sequences for a given ligand pose and protein scaffold. These synthetic data points could expand and improve the data used to train ML-SBDD methods. Systematic evaluations show their co-generated structures are often physically implausible and so inappropriate for downstream analyses, and that they do not sufficiently explore pocket sequence space. In doing these evaluations, I developed a series of benchmarks and tests that interrogate their performance.

In Chapter 5, I outline the initial steps taken to address the problems of ligand-pocket generation methods. Here, I introduce ScrewzFix, a guidance method to improve the speed of cofolding by trimming the protein. This faster inference

enables the development of Sparkz, a proof-of-concept hallucination framework for ligand-pocket design.

Finally, in Chapter 6, I summarise the results of this work and discuss future directions.

2

Robustly interrogating machine learning-based scoring functions: what are they learning?

Contents

2. Robustly interrogating machine learning-based scoring functions: what are they learning?

This chapter is based on work described in the following publication:

Guy Durant, Fergus Boyles, Kristian Birchall, Brian Marsden, and Charlotte M. Deane (2025). *Robustly interrogating machine learning-based scoring functions: what are they learning? Bioinformatics*, 41(2).

2.1 Preface

Binding affinity is a key property for a lead compound in early-stage drug discovery and is crucial to maintain while other properties are also optimised, such as ADMET (Gleeson 2008). MLBSFs learn the relationship between binding affinity and the static protein-ligand complex from existing labelled crystal structures datasets, such as PDBBind (Liu et al. 2014). These models promise to accelerate drug discovery by replacing the expensive and time-consuming experimental validation with accurate *in silico* predictions.

As outlined in Chapter 1, diverse and numerous architectures have been developed, but it remains unclear what architectural decisions, such as model type, data featurisation and model complexity, contribute to learning the underlying physics and which encourage the model to memorise dataset biases. These dataset biases are based on similarity patterns in the data that can be spurious.

Current benchmarks do not effectively discriminate between these two forms of performance. This motivated my development of simple baseline methods that were intentionally restricted from learning the underlying biophysics through the input featurisation of the data, and so were learning the "biases" of the data. These baseline models informed the design of novel benchmarks that would adversarially penalise memorisation. In this chapter, I describe how I retrained multiple models of different types with a fixed train-test split to control for training data differences. Furthermore, I explored the impact of structural accuracy in many contexts to understand how it impacts the accuracy of these methods.

By comparing different MLBSFs, I found that the tested models did not outperform these "bias" baselines. The new proposed benchmarks demonstrated

that all methods had poor performance when memorisation of data would not aid performance, indicating that all methods were likely not learning the underlying biophysics. Therefore, it appears that architectural decisions do not currently impact whether a ML model learns the underlying biophysics.

2.2 Introduction

As described in Chapter 1, predicting the binding affinity of a protein-ligand complex from its 3D structure has been extensively researched in the past decade (Meli et al. 2022). However, doing so for any protein-ligand complex accurately still poses a significant challenge in computational chemistry (Mobley et al. 2017). Accurately predicting binding affinity would aid in SBDD, where the chemical structure of a drug is designed based on the structure of its target, as it would enable the testing of design hypotheses in silico. One proposed methodology, scoring functions (described in Section 1.5.4) (Goodsell, David S and Morris, Garrett M and Olson, Arthur J 1996), which estimate binding affinity based on the features of a single protein-ligand complex structure, offers fast predictions and is suited for high throughput *in silico* hit identification and lead optimisation (Bissantz et al. 2000).

Docking software, such as AutoDock 4 (Morris et al. 2009), AutoDock Vina (Trott et al. 2010), Gold (Verdonk et al. 2003), and Glide (Friesner et al. 2004) (described in Section 1.5.5) commonly use scoring functions to predict the structure of the bound ligand (the pose), its binding affinity and its rank compared to other proposed poses. These scoring functions can use either molecular force fields (Huang et al. 2006a), statistical potentials (Gohlke et al. 2000) or linear combinations of empirical terms (Krammer et al. 2005). Advancements in ML have enabled the development of MLBSFs that outperform these other scoring functions in accuracy for predicting binding affinity. Initially, these scoring functions employed classical ML techniques, e.g. tree-based models, and simple features extracted from the protein-ligand complex structure (Ballester et al. 2010; Wang et al. 2017; Ballester et al. 2014; Li et al. 2015; Zilian et al. 2013; Durrant et al. 2011; Meli et al. 2021).

With the emergence of deep learning techniques, scoring functions based on the

2. Robustly interrogating machine learning-based scoring functions: what are they learning?

CNN architecture to predict the binding affinity only were built and trained on explicit, voxelised representations of the ligand-protein complex (Francoeur et al. 2020) (e.g. Pafnucy (Stepniewska-Dziubinska et al. 2018) and KDeep (Jiménez et al. 2018)). Newer deep learning methods, such as GNNs, represented atoms as nodes, bonds as edges, and used message-passing to pass feature vectors across the graphs to learn higher representations for predicting binding affinity (Moon et al. 2022; Karlov et al. 2020; Li et al. 2021; Scantlebury et al. 2023; Volkov et al. 2022). Despite the plethora of methods published, there is no clear consensus on which architectural decisions enable learning the underlying physics of protein-ligand binding instead of facilitating memorisation, given the small differences in performance observed between the methods on the standard benchmarks (Su et al. 2018; Carlson et al. 2016).

Most MLBSFs are trained on the PDBBind database (Wang et al. 2005), which consists of thousands of protein-ligand complex crystal structures with binding affinity data extracted from the literature. Complexes in CASF 2016 (Su et al. 2018), the most popular benchmark for scoring function performance, have very high similarity to data points within the standard training dataset (PDBBind), resulting in an over-optimistic measurement of accuracy as MLBSFs can memorise data similarity or "bias" instead of relevant biophysics (Scantlebury et al. 2023). This dataset and benchmark are described in further detail in Section 1.5.2. This has also been a problem in adjacent fields such as virtual screening, the classification of binders and non-binders to targets (Sieg et al. 2019; Wallach et al. 2018). Alternative methods of interrogation have been proposed by us and others; these include clustered cross-validation (Zhu et al. 2022), leave-cluster-out cross-validation (Kramer et al. 2010), time-splits (Volkov et al. 2022) and excluding training data similar to the test data (Scantlebury et al. 2023; Boyles et al. 2020). Unfortunately, due to the widespread use of the CASF 2016 benchmark for evaluating models, researchers can only compare their proposed model to others using that benchmark, exacerbating the problem of inadequate scoring function evaluation. Furthermore, MLBSFs are benchmarked and tested on accurate crystal structures, but they will

often be used for scoring predicted docked ligand poses against non-cognate or predicted structures in a real-world drug discovery setting. These noisy structures are likely to be less accurately predicted compared to the crystal structure, yet this impact has been explored in a limited manner for a few scoring functions (Boyles et al. 2022; Scardino et al. 2023; Wong et al. 2022; Shen et al. 2021; McNutt et al. 2021; Francoeur et al. 2020). Concerningly, others have demonstrated that models trained only on ligand and/or protein identities without explicitly including the interactions between them perform surprisingly well on CASF 2016 (Boyles et al. 2020; Volkov et al. 2022). It has been difficult so far to definitively prove what specific MLBSFs are learning due to the "black box" nature of many machine learning models and the lack of robust benchmarking in the literature. Nevertheless, it should be noted that learning biases is not inherently unhelpful and can be useful if models are used within the domain in which they have been trained, as discussed in Section 1.5.2. Prospective success is possible with MLBSFs, whether they have learnt bias or not (Hu et al. 2022).

In this Chapter, I present a platform for benchmarking and interrogating scoring function performance, ToolBoxSF. I explored the ability of these models to predict binding affinity values for a given single protein-ligand structure. First, I reimplemented a diverse set of MLBSFs: RFScore (Ballester et al. 2010), Pafnucy (Stepniewska-Dziubinska et al. 2018), PointVS (Scantlebury et al. 2023), SIGN (Li et al. 2021), and OnionNet-2 (Wang et al. 2021b), to use a consistent API and provide new tests and baseline models to interrogate their performance. By retraining them on different train-test splits, I was able to probe their differences consistently. I found that simple baseline models trained on only "dataset biases" and obfuscated from learning the underlying biophysics had competitive performance to the tested scoring functions in accuracy on these benchmarks. I also found that MLBSFs were still partially accurate in nonsensical scenarios, like scoring very high RMSD error poses, ligand docked into a random, wrong protein or complexes with deliberately induced clashes. These behaviours of these MLBSFs suggest they are also exploiting these dataset biases and have not learnt the underlying biophysics

of protein-ligand interactions from the training data's structures. The provided platform and results should enable researchers to fully and robustly interrogate their models and establish what could be the drivers of their performance.

2.3 Data and Methods

2.3.1 Training dataset

For consistency, models were trained on the popular PDBBind 2020 General, the most recent release at the time of the work (Liu et al. 2014). It consists of crystal structures of bound protein-ligand complexes with an associated binding affinity label (K_i, K_D or IC_{50}). Complexes that were unable to be processed by the latest versions of RDKit (2023.03.01) (Landrum 2023) or OpenBabel (3.1.1) (O'Boyle et al. 2011) were excluded, leaving 19,079 complexes for training and testing. Structures were prepared as described in 2.3.2 below, except that ligand coordinates were not recalculated. In this work, IC_{50}, K_i and K_D were treated as equivalent, a common approach in the field (Meli et al. 2022) despite the values not being strictly interchangeable (Kalliokoski et al. 2013). The pK for each compound was calculated by the following equation:

$$pK = -\log_{10}(K_i \text{ or } K_d \text{ or } IC_{50}) \tag{2.3.1.1}$$

2.3.2 Docking

Docking was performed using Smina, a fork of AutoDock Vina (Koes et al. 2013). The default parameters were used, except for "exhaustivenes" (set to 12) and"autobox_add" (set to 8Å). The protonation of the ligand and protein were kept consistent with those provided by PDBBind. Ligand MOL2 files from the PDBBind General 2020 dataset were converted to SDF format for consistency with the docked poses. Their 3D coordinates were recalculated using the ETKDG method from RDKit (Riniker et al. 2015) before docking to ensure the docking software was not able to use the crystal pose to influence its conformational search. Protein files had water molecules and any other non-protein atoms removed.

2.3.3 Benchmark preparation

Crystal Structure Benchmarks

To generate a benchmark where ligand bias cannot be used for accurate predictions, the 0 Ligand Bias benchmark, identical molecules were clustered, bound to different proteins by matching their InChI-Key (Pletnev et al. 2012). Only clusters whose mean pK value was within 6 and 7 pK units and whose variance was larger than 1 pK unit were retained, resulting in 365 complexes as a test set. These two final steps aimed to remove identical ligands with highly similar pK values and to ensure that predicting the mean of the clusters did not artificially increase the accuracy. For example, if two clusters had pK values concentrated around a low and a high value respectively, predicting the mean of each cluster would result in a high correlation between the predicted and true pK values across both clusters.

Peptides, defined as any entry in PDBBind with a ligand code with the letters "MER", were held out to create the Peptides Holdout (2574 complexes). This benchmark tested the scoring functions' ability to score peptides, having never been exposed to them in the training dataset. To be accurate on this benchmark, a scoring function must learn an understanding of biophysics that generalises from smaller molecules to peptides.

For the 2019 Holdout set, as done in (Volkov et al. 2022), any PDBBind data point with a crystal structure produced from 2019 or later was taken to form the test set (1511 complexes). This time split was designed to create a tougher test for scoring functions compared to CASF 2016, which was more representative of a drug discovery campaign.

Protein Structure Accuracy Benchmarks

To determine the effect of protein structure accuracy on performance, CASF 2016 ligands were redocked (Redocked) and crossdocked into protein conformations that were either bound but with the highest pocket similarity (CrossDocked (Best)), lowest pocket similarity (CrossDocked (Worst)), into apo structures (Apo), predicted AlphaFold 2 structures (AlphaFold 2) and a random wrong protein (Wrong Protein).

2. Robustly interrogating machine learning-based scoring functions: what are they learning?

For the Redocked set, ligands were docked back into their cognate structures using Smina. CASF-2016 includes five carefully selected protein–ligand complexes for each of 57 targets, featuring different ligands bound to the same protein (e.g., HIV protease). This setup introduces conformational variability within each protein, enabling cross-docking into alternative receptor structures within the test set. To generate the two classes of cross-docked structures, the pocket files, provided by PDBBind, were aligned using TM-Align (Zhang et al. 2005) in each set of 5 conformations and each alignment scored by TM-Score (Zhang et al. 2004). The highest TM-Score between a pocket and another pocket within the set was considered the best quality structure for cross-docking (Cross-docked (Best)), and the lowest TM-Score was considered the worst quality structure for cross-docking (Cross-docked (Worst)). The original ligand was docked at the site of the cognate ligand for the "best" and "worst" structures.

Apo structures for each cluster were identified using the search functionality on the PDB website. The PDB ID for the apo structure and its corresponding CASF 2016 PDB IDs are listed below. Only proteins that had no ligand bound in the active site and had 100% sequence identity for one of the five conformations in the set of 57 in CASF 2016 were considered. As the conformations in each protein family in CASF 2016 are not 100% sequence identical, 100% sequence identity between holo structures and their respective apo structure in CASF 2016 could not be guaranteed. Of the 57 sets in CASF 2016, only 46 had a suitable apo structure, so the final test set was only 230 complexes in size. For each complex, the apo structure was aligned to the original structure, and then the ligand was docked into the pocket of the apo structure. The PDB codes for the apo structures for each CASF 2016 are included in the Appendix (A.1.1).

For the Alphafold 2 version of CASF 2016, predicted structures using AlphaFold2 (Jumper et al. 2021) for monomers, and AlphafoldMultimer v2.1 (Evans et al. 2021) for proteins consisting of multiple polypeptides were generated. Predictions were run as described in the original publication (Jumper et al. 2021), including sequences from UniRef (Suzek et al. 2007) as well as BFD (Jumper et al. 2021) and Mgnify

(Mitchell et al. 2020). To emulate a realistic blind prediction scenario, templates were not included in the prediction, although, of course, a notable fraction of the targets will have been part of AlphaFold 2's training set. The ligand was then docked into the aligned AlphaFold structure. All CASF 2016 structures could be successfully predicted, except one: PDB:1YDR. Finally, for the Wrong Protein set, the ligand was docked into a randomly chosen protein from the CASF 2016 set that was not from the same protein family.

Protein Family Holdout Benchmarks

The Protein Family Out benchmarks were created to measure scoring function accuracy on specific protein families. The PDBBind dataset was clustered, using 90% sequence identity clusters from the PDB (PDB 2023) and took any cluster that had more than 100 data points as separate test sets to simulate the screening of a single protein target. The 100 data point size limit ensured there were sufficient data points to evaluate scoring function accuracy.

Docking Accuracy Benchmarks

To generate a diverse range of docking errors, the ligand was redocked back into the cognate structure of each protein-ligand complex of CASF 2016, 2019 Holdout and 0 Ligand Bias. The "autobox_add" parameter was increased to 20Å and "num_modes" to 1000 for Smina to increase the diversity of poses that could be generated. To generate more poses close in accuracy to the true pose, the crystal pose was minimised using the "minimize" option. Poses were binned by RMSD to the crystal pose using the following ranges: 0-1Å, 1-2Å, 2-4Å, 4-6Å, 6-8Å, 8-10Å, 10-15Å, 15-20Å, 20-25Å and 25-30Å. When available, the pose closest to the bin's mean was selected for each test set.

Clashes Benchmarks

To explore the models' sensitivity to steric clashes, a simplistic test for their understanding of biophysics, the crystal pose of the ligand for each complex from CASF 2016, 2019 Holdout and 0 Ligand Bias was progressively translated into the

protein 1Å at a time, ten times. To calculate a normal vector for the translation, the direction vector between the closest ligand atom and the closest protein atom in each protein–ligand complex was normalised.

2.3.4 Implementation of scoring functions and models

Existing machine learning-based scoring functions

To compare performance across a range of scoring functions, five popular and diverse models were selected from the literature: RFScore (Ballester et al. 2010), PointVS (Scantlebury et al. 2023), Pafnucy (Stepniewska-Dziubinska et al. 2018), SIGN (Li et al. 2021) and OnionNet-2 (Wang et al. 2021b). RFScore was one of the first methods to use machine learning to predict binding affinity. The model was trained using a Random Forest architecture and counts of protein and ligand elements that are within 12Å of each other as features. Pafnucy employs a CNN architecture and 3D voxelised representations of the protein-ligand complex. OnionNet-2 also utilises a CNN architecture with a 2D image of the counts of each specific amino acid-ligand atom interaction with differing thresholded distances. SIGN and PointVS both use GNNs with attention layers for the edges. PointVS is also pre-trained to classify pose accuracy within 2Å and uses this as a prior for its prediction of binding affinity. All differences between the original implementations and this work's modified implementations can be found in the Appendix (A.2.1).

Bias baseline models

Four separate baseline models were developed that represent models that can only learn "bias" in the dataset. The workflow of developing the baseline models is depicted in Figure 2.1. All baselines were developed using tree-based models with architecture and hyperparameters chosen by the FLAML package (Wang et al. 2021a) using five-fold cross-validation of the training dataset with CASF 2016 excluded. The LigandBias model is based on the simple QSAR-like model from Boyles et al. 2020. However, unlike QSAR methods, it is applicable to any protein and not a single protein like a standard QSAR model, where 1D and 2D descriptors

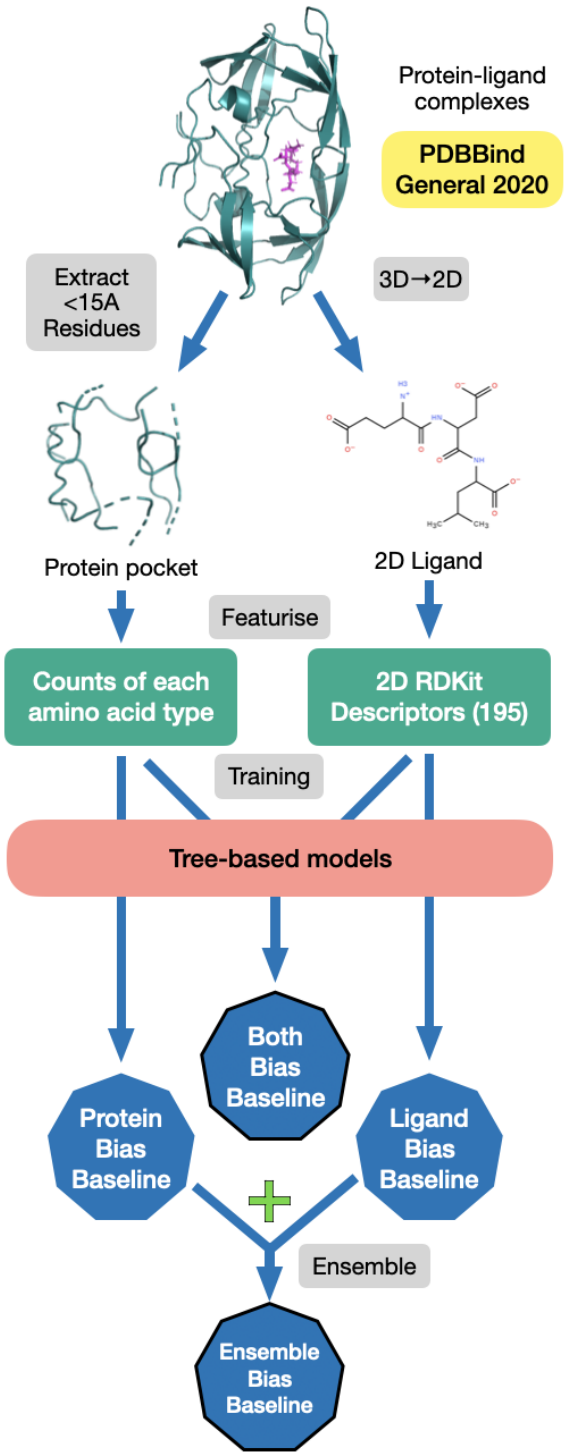


Figure 2.1: Training workflow of the baseline models.

.

from the RDKit package were calculated to featurise only the ligand. Any descriptor that produced NaN values or extremely large values for any example was excluded, leaving 195 features. The LigandBias model will always predict the same value of affinity for a ligand, no matter what protein it binds to, as it cannot see the protein. Therefore, it can only memorise ligand identity, and its performance can be ascribed to learning the ligand bias from the data.

The ProteinBias model used counts of each amino acid within the pocket as a feature vector. The protein pocket was defined as any amino acid that had an atom within 15Å of any ligand atom. The impact of this threshold on model performance in CASF 2016 is explored in the Appendix (A.2.2). These features give the ProteinBias model the identity of the amino acids but not proximity to each other or ligand atoms, and so severely limit the structural information in the features, and so can only memorise pocket identity or bias.

The predictions of the LigandBias and ProteinBias models were ensembled to give the EnsembleBias model, which is unable to see both biases at once. The final model, the BothBias model, concatenates features from both the ProteinBias and LigandBias models, which can learn from both sets of bias but is prevented from learning from the 3D structure, and so the underlying biophysics. The details on algorithms and hyperparameters for each baseline model are provided in the Appendix (A.2.3). All the test sets were also scored using Smina (Koes et al. 2013) as a baseline for the performance of a non-ML-based scoring function.

The data produced in this work and code for these models have been developed into an easy-to-use platform, called ToolBoxSF, to robustly compare to proposed models from the community and examine if they are learning more than bias. All models have been installed into separate Singularity containers to allow instant and easy use of the models for training or predictions. These code repositories are available on GitHub and as pre-built Singularity containers (https://github.com/guydurant/toolboxsf).

Method	CASF 2016	2019 Holdout	Peptides Holdout	0 Ligand Bias
LigandBias	0.76±.06	0.59±.03	0.23±.04	0.08±.11
ProteinBias	0.75±.07	0.59±.04	0.32±.04	**0.41±.10**
EnsembleBias	0.82±.04	0.68±.03	**0.37±.04**	0.27±.11
BothBias	**0.85±.03**	0.67±.03	0.35±.04	0.27±.12
Smina	0.59±.08	0.36±.04	0.19±.04	0.12±.10
RFScore	0.82±.04	0.64±.03	0.33±.04	0.24±.10
PointVS	0.79±.04	0.66±.03	**0.37±.04**	0.28±.10
Pafnucy	0.74±.06	0.60±.04	**0.37±.04**	0.17±.11
SIGN	0.82±.04	0.66±.03	0.34±.04	0.27±.10
OnionNet-2	0.82±.04	**0.70±.03**	0.36±.04	0.35±.10

Table 2.1: Pearson's R between predicted and true pK values for protein-ligand complexes for the baseline models (LigandBias, ProteinBias, EnsembleBias and BothBias), a non-ML-based scoring function (Smina) and five commonly used MLBSFs (RFScore, PointVS, Pafnucy, SIGN and OnionNet-2) on four benchmark datasets (CASF 2016, 2019 Holdout, Peptides Holdout and 0 Ligand Bias). See methods for further details of scoring functions and dataset creation. The highest values are in bold and underlined, with any value within the highest values' confidence intervals underlined. Error ranges represent the 95% confidence intervals from bootstrapped Pearson's R (N=10000)

2.3.5 Metrics

Scoring function accuracy was calculated between their predicted and true values using bootstrapped Pearson's R, R^2, and root mean squared error (RMSE) values, where data points were sampled with replacement 10000 times to produce 95% confidence intervals. Accuracy measured using Pearson's R is presented in the chapter, as is the most commonly used metric in this field; results for the other two metrics are provided throughout Appendix A.

2.4 Results

2.4.1 Existing Benchmarks

Evaluation of scoring function accuracy has typically been done using the CASF 2016 benchmark, so I first benchmarked each model on this set for Pearson's R, the commonly used metric used to assess scoring function accuracy (Table 2.1, CASF 2016), with results for R^2 and RMSE in the Appendix (A.3.1). However, the similarity between training (PDBBind) and test set (CASF 2016) makes this

an unsuitable benchmark for assessing MLBSF generalisability (Scantlebury et al. 2023). I compared five different models that featurise the protein-ligand complex differently: RFScore (Ballester et al. 2010), Pafnucy (Stepniewska-Dziubinska et al. 2018), PointVS (Scantlebury et al. 2023), SIGN (Li et al. 2021) and OnionNet-2 (Wang et al. 2021b). I retrained these scoring functions on the training sets and compared their performance against the baseline models, which were prevented from learning the underlying physics from the 3D structure through their bias features. The model trained on both protein and ligand bias features ("BothBias") had the highest values, although RFScore, SIGN and OnionNet-2 were within confidence intervals for all three metrics, demonstrating that learning biophysics from 3D information is not necessary for close to state-of-the-art performance on the standard CASF 2016 benchmark (Wang et al. 2021b). High performance on this benchmark has been shown not to be indicative of generalisability (Volkov et al. 2022; Scantlebury et al. 2023; Zhu et al. 2022), but this result goes one step further and demonstrates that even attempting to learn biophysics from structures of the protein-ligand complex provides no additional accuracy.

Volkov et al. proposed a time-split where PDBBind data points from 2019 and later were held out as a test set to account for this bias (Table 2.1 2019 Holdout). BothBias and EnsembleBias baselines are within confidence intervals for all metrics with OnionNet-2, the highest performing. This outcome indicates that a time-based split may not be suitable for demonstrating that a scoring function has learnt concepts of biophysics instead of dataset bias. Although both of these benchmarks have value in evaluating the accuracy of scoring functions, it is clear that other benchmarks are required to determine whether a model would be capable of generalising to novel protein or ligand families through learning the underlying biophysics.

2.4.2 New Proposed Benchmarks

Here, I propose two benchmarks which evaluate whether models are learning the training data's biases in different ways. The first utilises the difference between the

properties of peptide-protein complexes and ligand-protein complexes found within the PDBBind 2020 General dataset. I removed any peptide-containing complex as a hold-out set from the training dataset. Peptides are difficult to score due to their inherent flexibility and are often much larger than the other ligands in PDBBind (London et al. 2010). This makes it a difficult benchmark, but success would demonstrate that the models have learnt an understanding of biophysics, such as entropy and changes in solvation, from small molecules that generalise to peptides. I also explored the impact of scoring a subset of the peptides that had sizes in line with marketted peptide drugs in the Appendix (A.3.2). The results in Table 2.1 show that the BothBias baseline had performance within confidence intervals of the highest performing methods for Pearson's R. It should be noted that ProteinBias performed the most accurately in R^2 and RMSE, demonstrating the need for analysis of scoring function accuracy using more than the commonly used Pearson's R.

The second benchmark takes advantage of scoring functions tending to learn ligand-specific bias in that they are poor at differentiating between the same ligand bound to different proteins (Boyles et al. 2020). I identified identical ligands within PDBBind 2020 General that had existed two or more times in the dataset and filtered to ensure these identical ligands' mean and variance of pKs were centred but spread across the mean pK of the PDBBind dataset (i.e. the training dataset). These groups of identical ligands were then combined into a single set as the 0 Ligand Bias set. On this test set, ProteinBias had the highest performance, with OnionNet-2 and PointVS within confidence intervals for some of the metrics. Notably, BothBias was no longer within the confidence intervals of the best-performing. This demonstrates that ignoring the ligand is sufficient for the highest performance currently on this test set. Protein bias is helpful due to the similarity of protein pockets between the test and train sets (88% of test set pockets have the same Pfam ID as pockets in the train set (Finn et al. 2014)). Low performance across all models tested, across all metrics, indicates that this is a challenging benchmark. Furthermore, the similar performance of BothBias and EnsembleBias models in all benchmarks indicates

that BothBias is not reconstructing an understanding of 3D interactions using the 2D features, as EnsembleBias cannot with the same features but across two models. In the rest of this paper, I refer to only the results of BothBias for brevity due to these similar and therefore redundant results. I further analysed the predictive performance of just the molecular weight for each of the benchmarks, finding that only CASF 2016 showed any correlation (Appendix A.3.3). These benchmarks demonstrate that current scoring functions are not able to significantly outperform models trained to memorise only and prevented from learning the underlying physics. Therefore, these MLBSFs are both learning biases that do not generalise to this test set and learning little or nothing further.

2.4.3 Accuracy of MLBSFs on Protein Family Hold-Outs

The benchmarks investigated above do not reflect a realistic drug discovery scenario, as they measure accuracy across multiple different protein families instead of screening against a single protein target, which is the more common use of these scoring functions. To address this, I also created the Protein Family Out Benchmarks test sets to measure scoring function accuracy on specific well-represented protein families held out from the training data. These hold-out tests demonstrate the overall trend, as shown in Figure 2.2 with results for R^2 and RMSE in the Appendix (A.4), that the scoring functions do not vary greatly in the accuracy they achieve, with larger differences in the average ability between protein families than between model accuracy on the same family. The baseline models also follow this trend, suggesting that the reason for these differences is probably due to the protein families having different similarities to the training dataset rather than any more profound understanding of biophysics. The baselines do not always match the performance of existing methods, such as Thrombin and Trypsin, where PointVS is significantly higher. These results may be due to PointVS's exposure to these highly similar or identical complexes during its pre-training for pose classification.

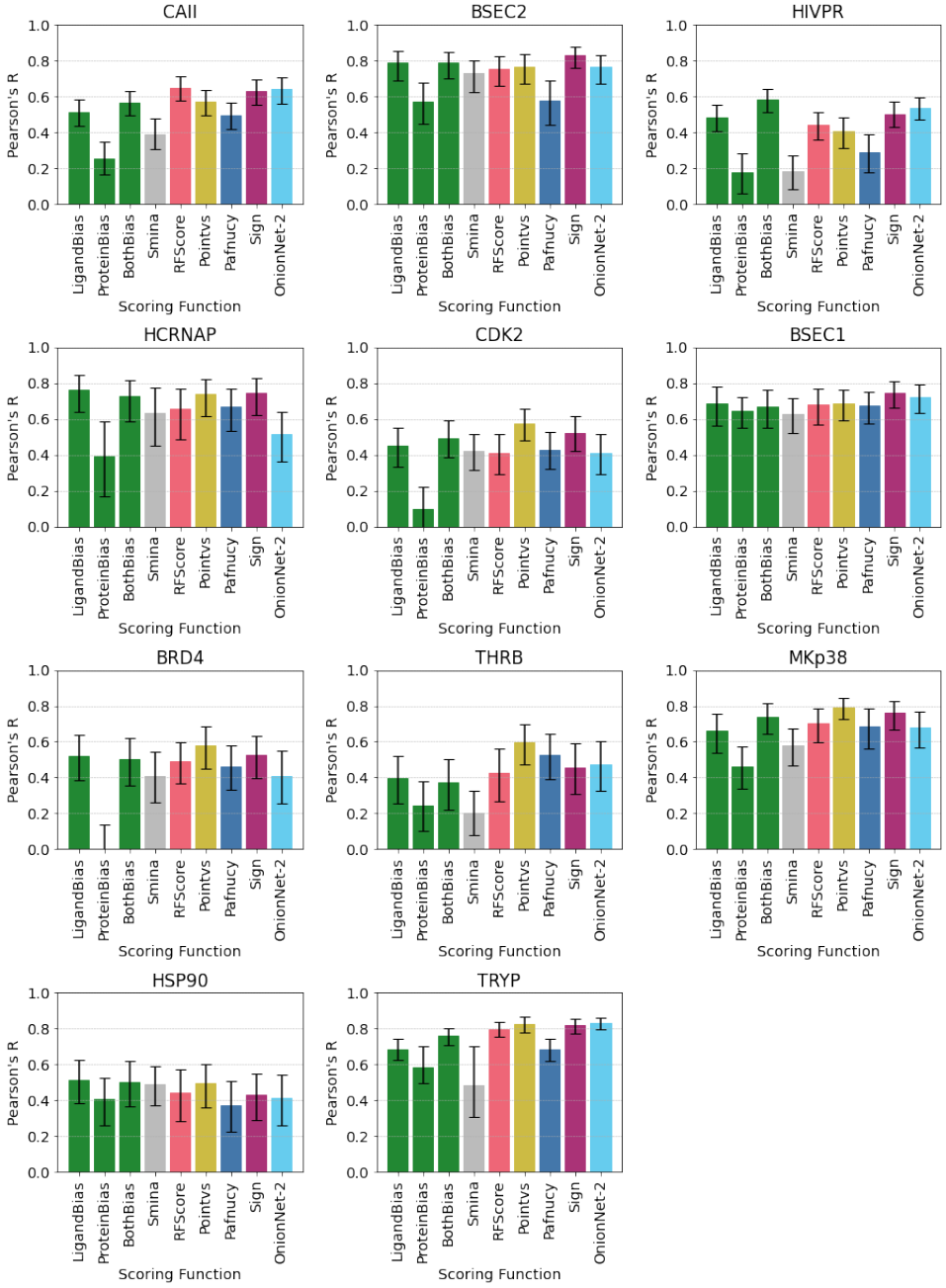


Figure 2.2: Pearson's R between predicted and true pK values for protein-ligand complexes for the baseline models (Ligand Bias, Protein Bias and Both Bias), a non-ML-based scoring function (Smina) and five commonly used MLBSFs (RFScore, PointVS, Pafnucy, SIGN and OnionNet-2) for eleven protein family hold-out clusters. These eleven families are Carbonic Anhydrase II (CAII), Beta-secretase (BSEC2), HIV protease (HIVPR), Hepatitis C Virus RNA-polymerase (HCRNAP), Cyclin-dependent kinase 2 (CKD2), Beta-secretase (BSEC1), Bromodomain-containing protein 4 (BRD4), Thrombin (THRB), MAP Kinase p28 (MKp38), Heat Shock Protein 90 (HSP90) and Trypsin (TRYP). Error bars represent the 95% confidence intervals from bootstrapped Pearson's R (N=10000).

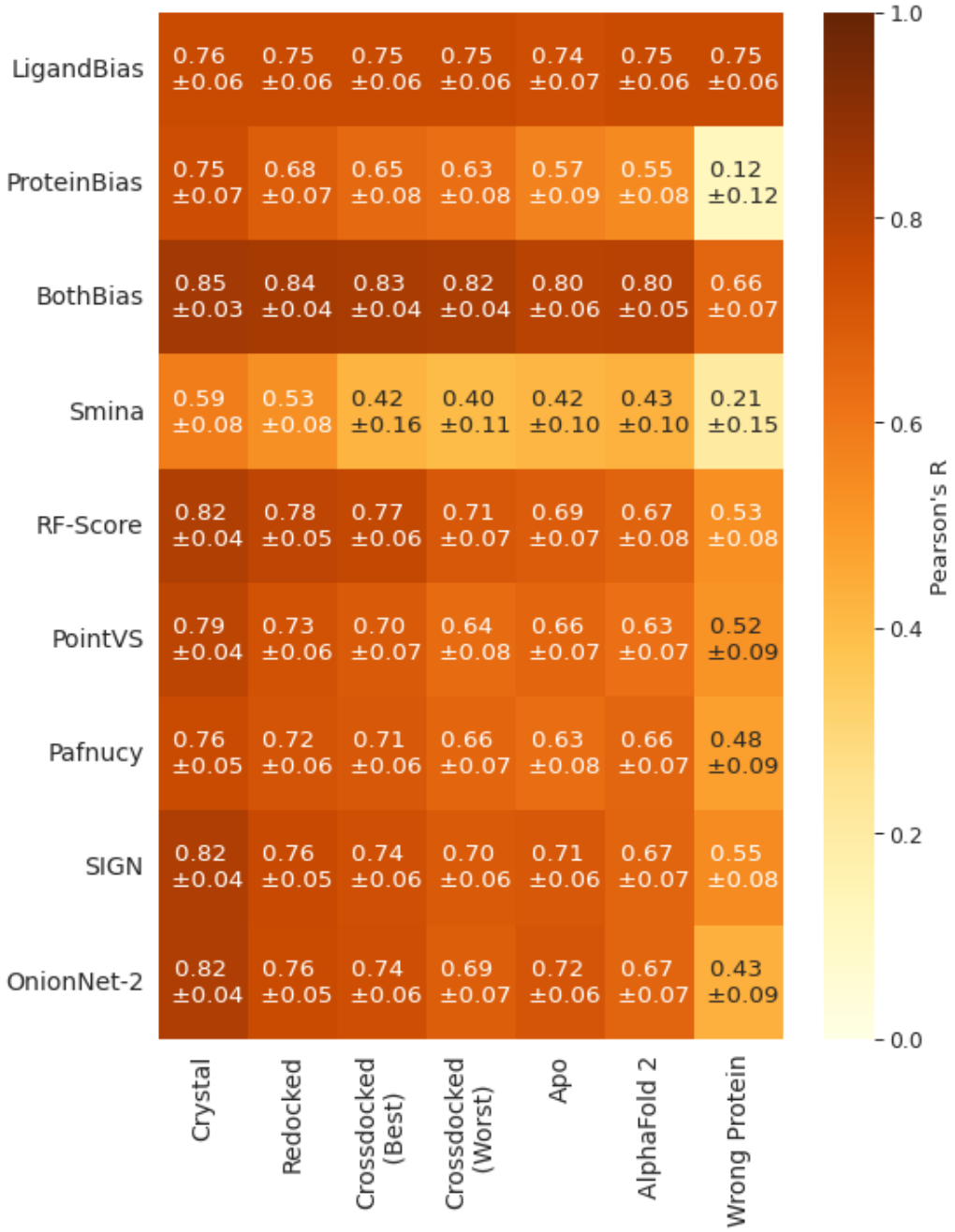


Figure 2.3: Pearson's R between predicted and true pK values for protein-ligand complexes for the baseline models (LigandBias, ProteinBias and BothBias), a non-ML-based scoring function (Smina) and five commonly used MLBSFs (RFScore, PointVS, Pafnucy, SIGN and OnionNet-2) on alternate CASF 2016 complex type test sets. Errors are the 95% confidence intervals from the bootstrapped Pearson's R (N=10000). Results using R^2 and RMSE are provided in the Appendix (A.5)

.

2. Robustly interrogating machine learning-based scoring functions: what are they learning?

2.4.4 Effect of Protein Structure Accuracy on Performance

One deficiency in using all the test sets employed above as benchmarks or held-out tests is that they only measure accuracy for scoring crystal structures. Typically, scoring functions are used to score docked poses against crystal or predicted structures that might not have an accurate active site conformation for the docked ligand. This introduces noise into the structure, as docking predictions may not find the specific interactions or recapture the true binding pose of the crystal structure. To explore the impact of this noise on accuracy, I created alternate docked versions of the CASF 2016 benchmark, which is made up of five structures, each bound to a different ligand, for each of 57 types of proteins (so 285 complexes total), and so contains alternate conformations for the same protein to dock into. I produced six test sets where I re-docked the ligand back into the cognate protein structure (Figure 2.3 Redocked), cross-docked it into a conformation most similar to its own (Figure 2.3 Crossdocked (Best)) and again into a conformation most dissimilar (Figure 2.3 Crossdocked (Worst)). I also docked the ligand into apo (unbound) structures (Figure 2.3 Apo), predicted Alphafold 2 structures (Figure 2.3 Alphafold 2), and a random protein from CASF 2016, not from its family, as a baseline (Figure 2.3 Wrong Pocket).

These increasingly noisy types of structure demonstrated decreased accuracy when scored by all scoring functions, as shown in Figure 2.3. The scoring functions were able to maintain a correlation with the true values even if the ligand was docked into a completely different protein, demonstrating a lower bound of accuracy caused by predictions being dominated by identifying the ligand rather than the nature of the complex. The BothBias model does not appear to be affected as much by the increasing noise, as its ligand features are not impacted by changes in conformation, and the number of amino acids in the protein pocket does not change significantly across the complex types. These results also suggest, as expected, that measuring performance on crystal structures provides an upper limit of the ability of scoring functions that is unlikely to be replicated if used in a drug discovery campaign (Brown et al. 2009).

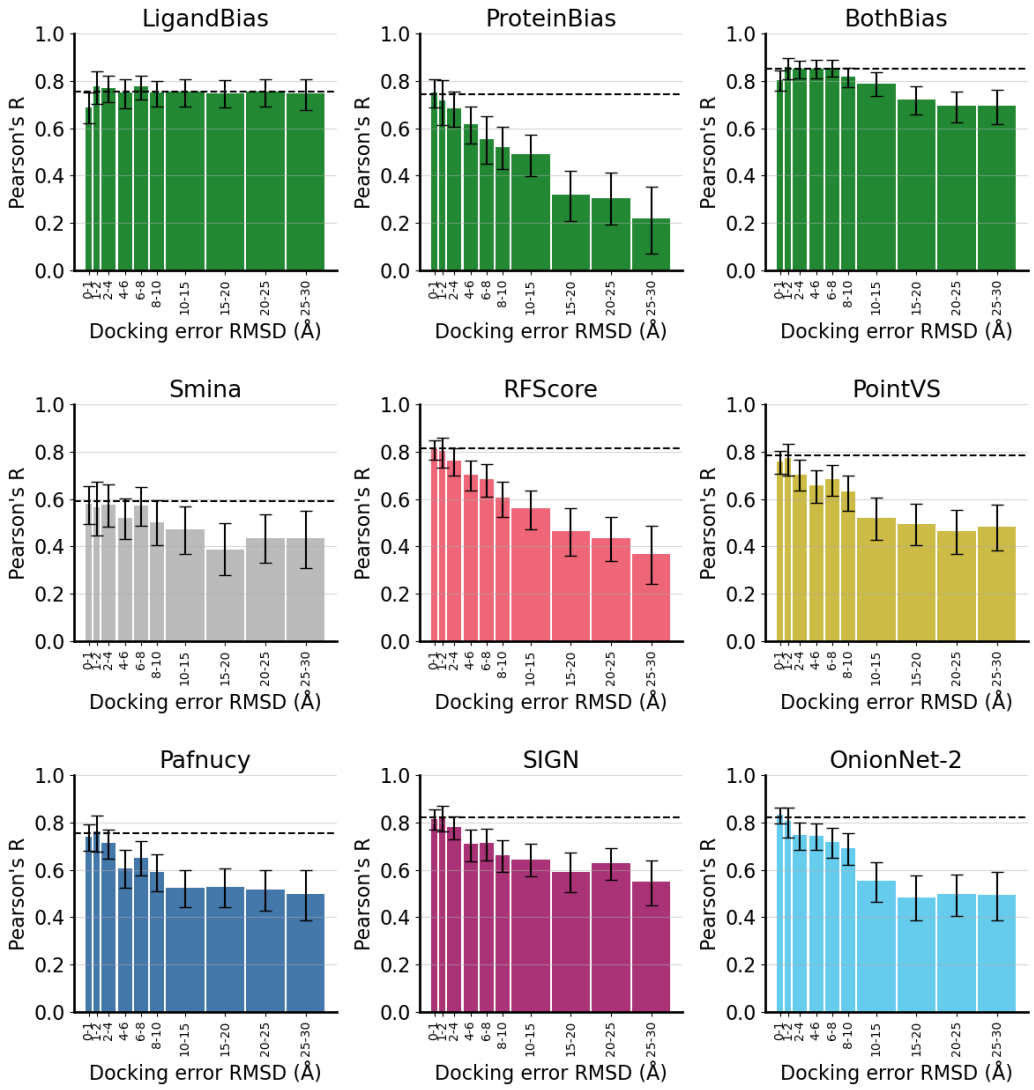


Figure 2.4: Pearson's R between predicted and true pK values for protein-ligand complexes for the baseline models (Ligand Bias, Protein Bias and Both Bias), a non-ML-based scoring function (Smina) and five commonly used MLBSFs (RFScore, PointVS, Pafnucy, SIGN and OnionNet-2) on different accuracy poses of CASF 2016 complexes. Accuracy on the crystal structures of CASF 2016 is shown as a dashed black line. Errors are the 95% confidence intervals from the bootstrapped Pearson's R (N=10000). Results using R^2 and RMSE are provided in the Appendix. (A.6)

.

2.4.5 Effect of Docking Accuracy on Performance

To measure the impact of docking accuracy, I considered a diverse set of poses for the CASF 2016 complexes, binned by RMSD. I binned the poses by RMSD first in small ranges (0-1Å and 1-2Å) and then increasingly larger bins with higher inaccuracy to produce 10 test sets. When high-accuracy poses were used, models

retained high predictive accuracy relative to scoring the crystal structures when scored by different scoring functions.

However, as docking error increased, correlation with true values decreased and ultimately plateaued at 10Å, except for RFScore, which continued to decline beyond this point. This plateauing occurs even for Smina, probably due to its ligand-size bias (Chang et al. 2010). Similar to the complex type tests, there was a lower bound for this decrease in performance even at extreme docking errors (25-30Å), where the ligand is no longer bound in the correct site, showing again that the models were relying on ligand bias to score protein-ligand complexes. I also explored this effect on 2019 Holdout and 0 Ligand Bias complexes, presented in the Appendix, and found the same trend (A.7, A.8). These results demonstrate that although docking accuracy is important for binding affinity prediction accuracy, bias is still a significant driver of scoring performance, as there is a correlation with true values for highly inaccurate poses.

2.4.6 Clashes

Finally, I investigated the behaviour of MLBSFs on clashing protein-ligand complexes by progressively translating the ligands into the proteins for each CASF 2016 complex. Although it is unlikely these scoring functions will be used to score these clashed structures in a drug discovery campaign, the energetically unfavourable overlap of ligand and protein structure is a simple example of biophysics that a model that has learnt the underlying biophysics would be responsive to. Therefore, an MLBSF that has learnt the underlying biophysics should fail to predict binding affinity with these increasingly severe clashes accurately. The MLBSFs displayed greater sensitivity to translation than the BothBias baseline model; however, most scoring functions displayed only a gradually decreasing performance as the clashes became increasingly severe, again indicating a lower bound (Figure 2.5). These results indicate that the scoring functions only recognise that the ligand is further from the binding site, rather than detecting the unphysical clashes with the protein.

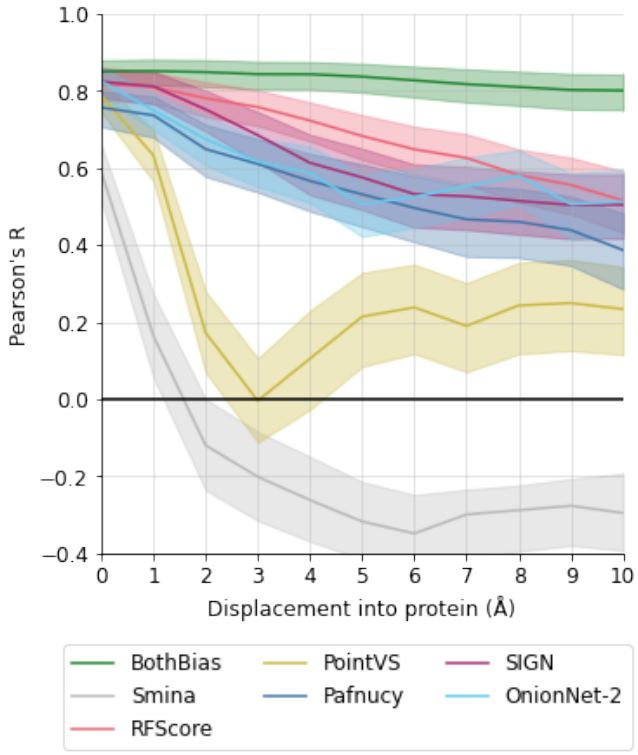


Figure 2.5: Pearson's R between predicted and true pK values for protein-ligand complexes for one baseline model (BothBias), a non-ML-based scoring function (Smina) and five commonly used MLBSFs (RFScore, PointVS, Pafnucy, SIGN and OnionNet-2) on progressively displaced ligands into the protein originally from CASF 2016 crystal structures. Errors are the 95% confidence intervals from the bootstrapped Pearson's R (N=10000). Results using R^2 and RMSE are provided in the Appendix (A.9)

.

The exceptions to these trends are Smina and PointVS, which are both co-trained or pre-trained for pose prediction and demonstrate higher sensitivity to clashes with low or no accuracy on complexes with significant clashes. Again, I also explored this effect on 2019 Holdout and 0 Ligand Bias complexes and found the same trends as for CASF 2016 (A.10, A.11) This suggests that considering pose quality in the training process, not just the binding affinity alone, provides scoring functions with the ability to discriminate between clashing, overlapping structures and true protein-ligand complex structures.

2. Robustly interrogating machine learning-based scoring functions: what are they learning?

2.5 Discussion

In this work, I have demonstrated that state-of-the-art performance on CASF 2016 can be achieved by baseline models using only protein and ligand bias. I have developed the 0 Ligand Bias and Peptide Holdout test sets, which either explicitly penalise learning ligand bias or require a greater understanding of biophysics, as tougher benchmarks and novel thresholds for improvement from the field. Five popular MLBSFs were equalled or outperformed by baseline models in these tests, indicating that the performance of these scoring functions may be due to learning dataset bias. I believe the baseline models offer a yardstick for the field, as if any proposed scoring function can significantly outperform them, they will have learned more than simple dataset bias.

I examined the effect of noise in the 3D structure of the protein-ligand complex on scoring function performance. Using inaccurate active site conformations or docked poses introduced structural noise, which reduced accuracy in proportion to the level of noise. However, both decreases in correlation to true values had a lower bound, showing indifference to the 3D structure input and instead relying on recognising the identity of the ligand.

A further proof that these models are not necessarily learning relevant biophysics is their insensitivity to serious steric clashes between protein and ligands. Translation of the crystal pose into the surface of the protein resulted in a gradual decrease in performance, indicating that the scoring functions were only able to recognise that the ligand was further from its true location. The exceptions to this trend, PointVS and Smina, were either pre-trained or developed for pose classification or ranking, respectively. These exceptions suggest that scoring functions trained to predict only binding affinity do not learn how sensible a pose is, whilst co-training for another task, such as pose classification, forces it to appreciate clashes. However, it must be noted that Smina never outperformed any of these MLBSFs in accuracy on any benchmark.

Overall, this work has provided a framework for benchmarking MLBSFs and created baseline models that equal existing scoring function accuracy and has

provided train-test splits that can help identify if proposed models have learnt more than this simple dataset bias. For the field to progress, it will be necessary to design and train models in such a way that they cannot achieve apparent success on benchmarks simply by learning dataset biases. I propose that by training on data that penalises memorisation, such as more of the data consisting of identical ligands bound to different proteins found in 0 Ligand Bias, these models could learn the underlying biophysics. Such data does exist in binding databases such as ChEMBL (Gaulton et al. 2012) and BindingDB (Gilson et al. 2016) but requires high-quality docking to produce structures for training these MLBSFs. To create sufficient training data for this, we need to have more reliable docking scoring to triage which poses might be accurate enough. In Chapter 3, I describe PoseTriager, a tool with improved calibration and accuracy for scoring docked poses. For researchers to prove their proposed scoring functions have learnt more than dataset bias, I have presented rigorous tests and baseline models that can elucidate whether they have learnt the underlying biophysics. All code and dataset splits can be accessed here: (https://github.com/guydurant/toolboxsf)

3

PoseTriager: improving pose classification robustness using data augmentation

Contents

3.1 Preface

Small molecule docking is used to predict the conformation of a small molecule bound to a macromolecular target. The proposed structure can be used for further analysis, such as binding affinity prediction, identifying SARs and simulation

studies. As detailed in the introduction, physics-based docking methods that optimise functions that approximate known physics have been shown to generalise with limited accuracy (Buttenschoen et al. 2024; Škrinjar et al. 2025). ML-based methods are more accurate in the data distribution they are trained on and can explore larger search spaces more efficiently, although they struggle to produce physically plausible poses (Buttenschoen et al. 2024). Both types of methods tend to produce multiple possible poses and rank them to provide the user with the best possible pose, with each method having its own ranking method.

The reliance on independent ranking methods results in docking methods not necessarily being interchangeable or combinable. Furthermore, these ranking methods, when employing ML, are often trained on the docking software's outputs to predict the confidence or score for a pose. This potentially biases the ranking methodology to predict pose quality based on the pathologies and preferences of a particular docking software instead of the provided 3D structure. Each docking method has different strengths and drawbacks across protein and chemical space (Buttenschoen et al. 2024; Škrinjar et al. 2025). By combining all the outputs and accurately ranking them at once, the performance of docking could be improved. Advancements in docking would also enable the building of more robust datasets for training MLBSFs proposed at the end of Chapter 1 to supplement those consisting only of experimentally determined structures.

This dataset development motivated the work in this chapter, where I explore the generalisability of pose classifiers trained only on Smina (Koes et al. 2013) outputs to score poses noised adversarially and from other docking methods (Gold (Verdonk et al. 2003) and DiffDock (Corso et al. 2022)). Further, I built Augmented2020, a set of noised Smina poses that were adversarially out of distribution, in an attempt to improve generalisation. The results on docking benchmarks demonstrated that training on this augmented data often did not improve accuracy and did not help with docking generalisability, with the only exception being when docking into apo structures using DiffDock. This suggests that addressing this lack of robustness to

distribution shift or noise is not a major driver for improving pose classification accuracy across different docking methods.

3.2 Introduction

Protein-ligand docking is a key tool in drug discovery, used to predict how small molecules bind to protein targets, which enables the generation of structural hypotheses about the biophysics of their interactions (described in detail in Section 1.5.5). Older docking methods have relied on biophysical approximations such as molecular force fields (Huang et al. 2006b), statistical potentials (Gohlke et al. 2000), or linear combinations of empirical terms (Krammer et al. 2005). Recent applications of ML, such as diffusion (Corso et al. 2022), transformers (Stärk et al. 2022) and flow matching (Lipman et al. 2022) methods, have aimed to revolutionise this process by learning directly from experimental data, such as crystal structures from the PDB (Burley et al. 2017), enabling more accurate pose prediction. These models are trained to match the experimentally observed ligand conformations or pose and enable wider applications of docking, such as pocket-defined docking methods (Plainer et al. 2023), blind docking against the whole protein surface (Corso et al. 2022; Morehead et al. 2025; Stärk et al. 2022), and co-folding, the joint prediction of ligand and protein structure (Krishna et al. 2024; Abramson et al. 2024; Wohlwend et al. 2025). However, docking methods have been found to produce physically implausible poses (Buttenschoen et al. 2024), not recapitulate the true interactions of the poses (Errington et al. 2025), and not to generalise novel pockets or proteins (Škrinjar et al. 2025; Buttenschoen et al. 2024). These shortcomings underscore the need for further methodological improvements in the field.

One key area where progress is required is the accurate scoring of the generated poses. Docking software will often explore the landscape of poses, generating many examples, with a scoring or "confidence" model ranking the poses. The top-ranked pose(s) are typically selected for further analysis and play a key role in guiding subsequent drug discovery efforts. However, to ensure that a researcher can reliably

interpret these structures, the scoring function must provide a meaningful estimate of pose quality. Physics-based methods score poses using energy functions, but these often consist of linear combinations of empirical terms whose predictions correlate with ligand size (Chang et al. 2010) and so can be poorly calibrated for pose scoring across different protein-ligand complexes. Deep learning methods lack an equivalent energy function for scoring, necessitating the use of a separate confidence score to rank their outputs. Outputs can be ranked by an external model, such as RTMScore (Shen et al. 2022), or by training a model alongside the generative model that learns to predict the pose error. Pose confidence or scoring can be measured by many different error metrics, such as RMSD from the ground truth pose, the distogram of difference in positions from the ground truth, the interaction similarity (Errington et al. 2025) or the local distance difference test (LDDT) (Mariani et al. 2013).

Examples of methods in this area include the DiffDock confidence model, which is trained to predict the RMSD error as a continuous value (Corso et al. 2022). AlphaFold-3 (Abramson et al. 2024), Boltz-1 (Wohlwend et al. 2025), and Chai-1 (Chai-Discovery-Team et al. 2024), predict multiple confidence metrics such as pLDDT, predicted atomic error (pAE) and predicted TMScore (pTM) (Zhang et al. 2005). Developers validate these confidence models by ranking performance alone, rather than discrimination accuracy, the determining of whether the top-ranked pose is correct. Additionally, most confidence models are tightly coupled to their respective generative methods and cannot be applied to other docking tools. A scoring model that is both well-calibrated and docking-tool agnostic would allow for more reliable pose triaging, enabling users to combine outputs from multiple docking methods and select the best pose regardless of origin. Such a method could enhance the usefulness of structure-based drug discovery pipelines.

To develop a more general or even a docking-specific pose classifier requires training data that includes both positive and negative examples of poses. Typically, docking pose scorers are trained on the outputs of the docking software themselves, with the poses labelled based on their accuracy, measured by different metrics. Training on docking outputs rather than completely random poses ensures the

model learns to distinguish between correct poses and incorrect yet plausible decoys that reflect the biases and search space of docking. Without exposure to these realistic decoys, a classifier is unlikely to generalise to the types of errors produced by docking algorithms in practice. However, this data generation approach introduces bias because it only reflects the types of poses that a specific docking software can produce. It fails to account for cases where the docking software cannot accurately model specific protein-ligand complexes. As a result, the classifier is only exposed to a narrow distribution of poses and may struggle to generalise, particularly when evaluating poses generated by other docking tools, which can appear adversarially out-of-distribution. Moreover, these datasets used for training are often not reproducible because the training data is not made publicly available, such as DiffDock (Corso et al. 2022) and Boltz-1x (Wohlwend et al. 2025).

A well-known and widely used source of training data for pose classification comes from the Koes group: Redocked2020, in which poses are docked into their cognate structures using Smina, and CrossDocked2020, in which noisier protein conformations are used in addition for cross-docking (Francoeur et al. 2020). These datasets were used to train the GNINA scoring function (McNutt et al. 2021). GNINA is a CNN co-trained to classify whether poses are within 2Å RMSD of the crystal structure and to predict binding affinity. GNINA is primarily intended for use on poses generated by AutoDock Vina (Eberhardt et al. 2021) or its fork Smina (Koes et al. 2013), both physics-based docking functions.

Here, I use these datasets as a case study to examine how well pose classifiers trained on docking tool–specific data generalise to poses generated by other docking software. Specifically, I show that simple EGNN models trained on Redocked2020 and CrossDocked2020 may perform less reliably when evaluating adversarial noisy poses outside the pose distribution produced by Smina. To address this, I developed PoseFoundry, a pipeline for generating accurate, physically plausible poses with controlled noise. Using this pipeline, I developed a new dataset, Augmented2020, based on poses from Smina but augmented with noise. This augmentation showed an increase in model robustness to noise, but did not result in a real increase

in accuracy in docking classification, except in docking applications involving very high noise levels. Code and model weights are available at the following links: PoseFoundry: https://github.com/guydurant/pose_foundry and PoseTriager: https://github.com/guydurant/posetriager.

3.3 Data and Methods

3.3.1 Benchmark Data

Posebusters Benchmark Set

The PoseBusters benchmark set is a popular time-holdout test split of the PDB for measuring ML docking accuracy (Buttenschoen et al. 2024). It has a cutoff of September 30, 2019, meaning it has no chronological overlap with the Redocked2020 and CrossDocked2020 datasets (latest deposition: March 17, 2019). Additionally, I stratified the dataset into three subsets by sequence similarity (0-30%, 30-95%, 95-100%) to the training dataset as in the original publication. The PDB codes for the splits are provided in the Appendix (B.1). These subsets enable analysis of how train–test similarity affects pose classification accuracy.

3.3.2 Docking

Smina

The input receptor (protein and associated cofactor atoms) and ligand files were protonated by Smina (Koes et al. 2013) using the internal OpenBabel library (O'Boyle et al. 2011). Waters were removed from the receptor files. The crystal ligand was used as the pose to create a box around using the "autobox_add" = 8Å, "exhaustiveness" = 32, "num_modes" = 40, "energy_range" = 200 and "min_rmsd_filter" = 1.

Gold

Waters were removed from receptor files, which were then protonated using the CCDC Python API (Sykes et al. 2024). For ligand files, standardisation of aromatic and delocalised bonds and then protonation was performed using the CCDC Python

API. Both receptor and ligand files were saved in MOL2 format. The fitness function chosen was "plp". The centroid of the crystal ligand's atomic coordinates was used as the centre of geometry for a spherical docking search space, with a radius of 25Å.

DiffDock

The latest release of DiffDock (Corso et al. 2022) (https://github.com/gcorso/DiffDock) was used for blind docking, where no pocket needs to be specified and the docking search space is over the whole protein. All default parameters were used except that 40 poses were predicted per protein–ligand complex instead of ten.

Boltz-1x Apo Docking

To generate apo receptor structures that are likely to be challenging to dock into, I used the Boltz-1x cofolding model (Wohlwend et al. 2025). This model uses guidance and steering to improve its physical plausibility to generate or "co-fold" structures of the proteins, with organic and inorganic cofactors bound. To limit the computational expense of predicting the entire complexes, I only co-folded any protein chain within 10Å of the crystal pose of interest. Furthermore, I included any inorganic or organic cofactor that was within 5Å of any of these protein chains that was not the ligand of interest for docking. MSAs for protein chains were generated using the MMSeqs2 server (Steinegger et al. 2017), as part of the Boltz package. I predicted ten complexes per protein-ligand complex and took the highest "confidence" score, a weighted average of the different confidence metrics of Boltz-1x, complex as the single apo structure for docking.

Pose post-processing

All poses generated through docking were evaluated using the PoseBusters software, and poses that failed any test, except whether the pose's RMSD from the ground truth pose was below 2Å, were excluded. This ensured that classification performance reflected discrimination by RMSD rather than being confounded by physical plausibility, which can be detected using thresholds. Statistics for the poses outputted by different software for different docking benchmarks are provided in

Table 3.1. The differences in the number of physically valid (measured by PBValidity using the PoseBusters package (Buttenschoen et al. 2024)) & RMSD <2Å poses can be attributed to both the ability to generate physically plausible poses and the diversity of poses each method generates. Gold produces the highest total as it does not enforce high diversity of poses like Vina, whilst DiffDock struggles to generate physically plausible poses (Buttenschoen et al. 2024).

Table 3.1: Docking statistics for Redocked and Apo docking PoseBusters benchmarks using Smina, Gold, and DiffDock. Note: Gold was unable to dock 44 apo structures due to errors in protein preparation.

Metric	Condition	Smina	Gold	DiffDock
Total poses	Redocked	11,921	12,280	12,160
	Apo	11,594	13,898	11,840
Total PDBs docked	Redocked	304	307	308
	Apo	293	251	296
Total PBValid poses	Redocked	11,121	11,271	2,788
	Apo	10,722	8,882	1,435
PBValid & RMSD < 2Å poses	Redocked	609	5,064	2,090
	Apo	223	1,368	644

3.3.3 Models

PoseTriager

I reimplemented the EGNN model, PointVS, developed by previous members of my group (Scantlebury et al. 2023). This architecture was chosen over the CNN architecture of GNINA to avoid the need for expensive data augmentation to learn invariance. Furthermore, the architecture had been optimised for the Redocked2020 dataset. I reduced the number of layers from 48 to 4 to avoid oversmoothing of the node features in the graph (Li et al. 2018). The simple graph structure and featurisation of the nodes and edges were kept the same as in PointVS. All protein atoms within 6Å of the ligand are added as nodes, along with all heavy atom ligand atoms, into the graph, and edges are built between nodes within 10Å of each other; therefore, covalent bonds were not explicitly modelled. Node features are a one-hot

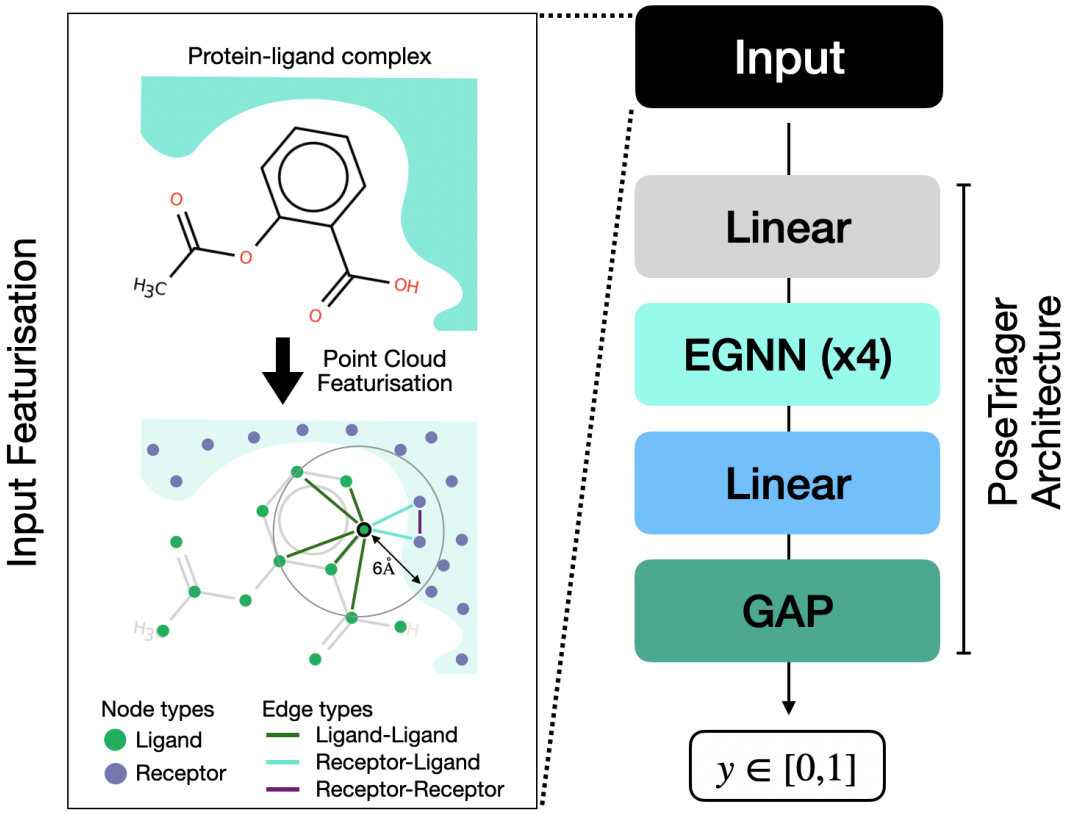


Figure 3.1: PoseTriager model architecture and point cloud featurisation. Input point clouds are passed through a single linear layer, 4 EGNN layers, a final linear layer and a global average pooling (GAP) layer to output a single probability for the input protein-ligand complex.

encoding of the Smina atomtypes (Koes et al. 2013), and edge features are also kept as one-hot encodings denoting ligand-ligand, ligand-receptor and receptor-receptor types. A depiction of the model architecture and graph featurisation is included in Figure 3.1. For model training, I divided the training sets into train and validation sets using a 30% sequence similarity clustering from the PDB (PDB 2023), resulting in a 90:10 split. I trained the classification models over 40 epochs using a binary cross-entropy loss and retained the model weights with the lowest mean-squared error on the validation set, to optimise for calibration accuracy. The learning rate was set to $6 * 10^{-4}$ with a weight decay of $1 * 10^{-5}$. To generate repeats, five separate models were trained for a given training-validation set with random seeds.

3.3.4 Pose Design with PoseFoundry

To generate poses in a more unbiased manner, instead of sampling from the outputs of docking software, I implemented a novel pose sampling methodology. The input pose, which could be from a crystal structure or a docked pose, is used as the starting structure. I recalculated the pose coordinates using the ConstrainEmbed function in RDKit (Landrum 2023) to generate conformers, with realistic bonds and bond angles, and atomic positions close to the input molecule. This function updates a distance bounds matrix based on the interatomic distances of the input molecule, produces conformers based on these constraints, and then further optimises with the forcefield. This step created highly accurate poses relative to the input, but they do not inherit the possible energetically unfavourable bond lengths and bond angles, such as those sometimes found in crystal ligand poses (Francoeur et al. 2020).

I used RDKit (Landrum 2023) and SciPy (Virtanen et al. 2020) to perform Latin hypercube sampling across the ligand's roto-translational and torsional search space. The bounds of the search space for translation were 1.0Å from the starting conformer, for rotation, 0.1 radians in each direction in polar space around the original pose and for each torsional angle, $\pi/3$. If the ligand had greater than five torsional angles, a random number between five and the maximum number of rotatable bonds was picked for each epoch of sampling. These limitations on the search space substantially improved the efficiency of finding a pose that fits a specified criterion, as increasing the search space led to intractable search times. 10,000 samples for each epoch were searched by PoseFoundry, with the number of epochs dependent on the type and difficulty of the pose required. Poses were then measured for whether they passed the selection criteria hierarchically based on speed of calculation, the fastest being first; in this case, RMSD was used as the first discriminating criterion. If the pose passed all specified criteria, it was saved; otherwise, it was discarded if it failed one or more criteria. I depict this process in Figure 3.2. I was able to specify samples that passed all PoseBusters checks and had specific RMSDs within ranges, for example, 1-2Å RMSD. I was also able to sample poses that failed specific PoseBusters checks. These three were: Protein-Ligand

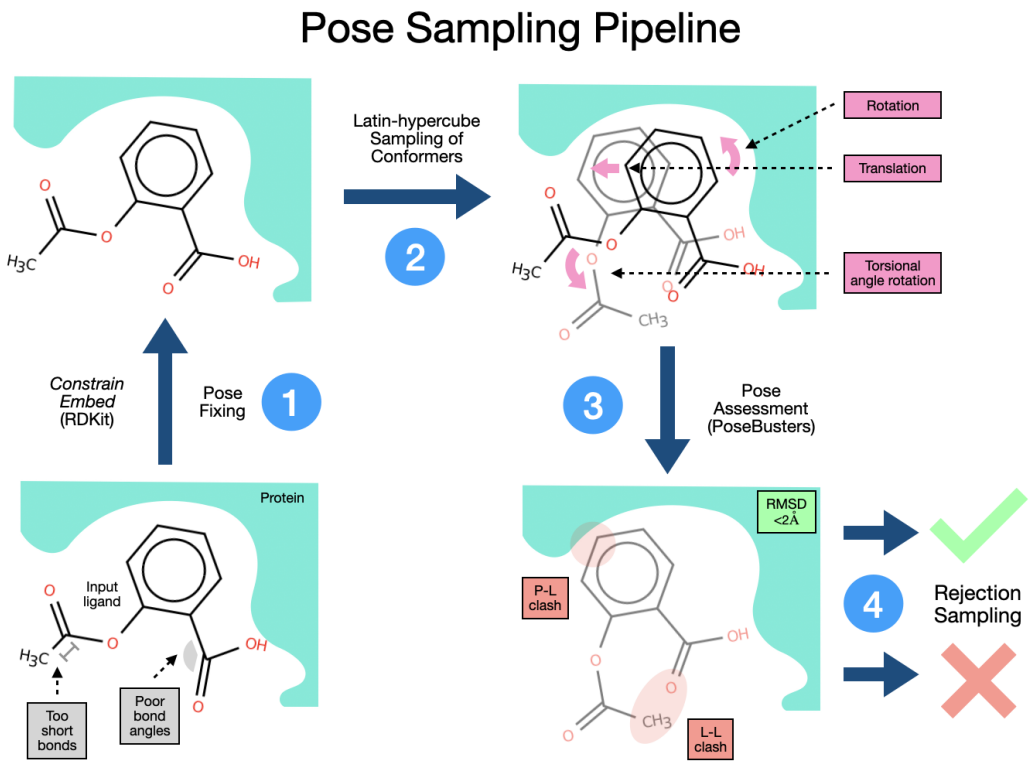


Figure 3.2: Full pose sampling pipeline for the generation of unbiased poses with specified criteria. 1) Poses are corrected for incorrect geometries, such as too short bond lengths or incorrect bond angles. 2) Translations, rotations and torsional angle rotations are sampled randomly using Latin-hypercube sampling. 3) Proposed poses are assessed for physical plausibility, such as protein-ligand clashes (P-L clash) and ligand-ligand clashes (L-L clash). They are also assessed for the accuracy of the pose, in this case, the root mean squared deviation (RMSD) of atomic positions to the input pose. 4) Finally, the pose is either accepted or rejected based on whether it has passed all the design criteria or not.

.

Clash, which specifies that a ligand or receptor (protein or non-protein) atom cannot be less than 75% of the combined Van der Waals (VdW) radius of the atoms. Next, the Protein-Ligand Volume Overlap specifies that no more than 5% of the volume of the ligand may overlap with the receptors. Finally, the internal clash checks that the ligand does not clash with itself, also with a lower limit of 70% of the combined VdW radius. I also included the Energy Ratio from PoseBusters, which specifies that the relative energy of a pose using the UFF forcefield (Casewit et al. 1992) in RDKit should not exceed a ratio of 100 times that of the average of 50 randomly generated poses using the ETKDG method (Wang et al. 2020). These

two PoseBusters checks were combined, as I found it was rare for a conformer to have an internal clash without producing too high relative energy conformations.

3.3.5 Training datasets

Redocked2020 and CrossDocked2020

The entire Redocked2020 (n=1.5m) and CrossDocked2020 (n=35.6m) datasets were downloaded from the following link: http://bits.csb.pitt.edu/files/crossdock2020/. The original datasets were generated by docking the 20,651 PDBs from the Pocketome database (Kufareva et al. 2012), using Smina, into their cognate structures (Redocked2020) and additionally into similar holo structures, clustered by Pocketome (CrossDocked2020). The datasets were also supplemented with actives by minimising the crystal pose in the UFF forcefield (Casewit et al. 1992) (to remove unfavourable molecular geometries) and then in the Smina forcefield. The datasets development is further detailed in Section 1.5.2. The authors prepared the receptor files by removing all waters and any cofactors that were not metal ions. Due to time and computational constraints, I used the subsampled version of CrossDocked2020, which limited the 35.6m poses of the set to just 1.6m by oversampling actives for each PDB code to 10 and undersampling decoys to 20.

Augmented2020

The Augmented2020 set was developed by docking ligands from the 20651 protein-ligand crystal structures used in the above datasets, to allow for fair comparisons. I also docked ligands into their respective cognate protein structures using Smina, as described in the docking section for Redocked2020. However, I applied the pose sampling method, PoseFoundry, described above, to augment the poses with noise for both the inaccurate docks produced by Smina and the crystal pose, thereby generating accurate, balanced active and decoy poses for each protein-ligand complex. I sampled five actives for each PDB and 15 re-docked poses whose resultant noisy poses were not within 2Å RMSD of the ground truth but were within 2Å RMSD of the initial docked pose. I further generated three PoseBusters checks-failing

(PBInvalid) poses by setting the criteria to specify specific failures described above, whilst still maintaining <2Å RMSD accuracy.

If specific PBInvalid poses could not be created, for example, if a ligand had no rotatable bonds and so could not be made to clash internally, I used another protein-ligand clashing pose, as this was the easiest type of pose for PoseFoundry to produce. All inorganic and organic cofactors were kept in the receptor structure, unlike the Redocked2020 and CrossDocked2020 datasets, which retained only metal ions. I limited the training set to use the PDBs that could have all types of poses generated for the above-mentioned types to ensure dataset balance, limiting the dataset to 19247 of the 20651 PDBs in the Redocked2020 and Crossdocked2020 sets.

3.3.6 Metrics

AUROC Area under the Receiver Operating Characteristic curve (AUROC) is defined as the area under the curve obtained by plotting $\mathrm{TPR}(t)$ versus $\mathrm{FPR}(t)$ as the decision threshold t varies:

$$\mathrm{AUROC} = \int_0^1 \mathrm{TPR}(\mathrm{FPR})\, d(\mathrm{FPR}),$$

where

$$\mathrm{TPR}(t) = \frac{\mathrm{TP}(t)}{\mathrm{TP}(t) + \mathrm{FN}(t)}, \quad \mathrm{FPR}(t) = \frac{\mathrm{FP}(t)}{\mathrm{FP}(t) + \mathrm{TN}(t)}.$$

Here, TP, FP, FN, and TN are true positives, false positives, false negatives, and true negatives, respectively. TPR is the true positive rate, and FPR is the false positive rate. The integral is computed numerically using the trapezoidal rule.

AUPRC Area under the Precision–Recall curve (AUPRC) is defined as the area under the curve obtained by plotting $\mathrm{Precision}(t)$ versus $\mathrm{Recall}(t)$ as the decision threshold t varies:

$$\mathrm{AUPRC} = \int_0^1 \mathrm{Precision}(\mathrm{Recall})\, d(\mathrm{Recall}),$$

where

$$\mathrm{Precision}(t) = \frac{\mathrm{TP}(t)}{\mathrm{TP}(t) + \mathrm{FP}(t)}, \quad \mathrm{Recall}(t) = \frac{\mathrm{TP}(t)}{\mathrm{TP}(t) + \mathrm{FN}(t)}.$$

Here, TP, FP, and FN are true positives, false positives, and false negatives, respectively. The integral is also computed numerically using the trapezoidal rule.

MCC Matthews Correlation Coefficient (MCC) is a balanced measure of binary classification quality:

$$\text{MCC} = \frac{\text{TP} \cdot \text{TN} - \text{FP} \cdot \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}.$$

Where TP, FP, TN, and FN are true positives, false positives, true negatives, and false negatives, respectively.

Top-1 Accuracy The proportion of test cases where the top-ranked pose (ranked by predicted score $\hat{p}_i$) has root mean squared deviation (RMSD) below a success threshold (e.g. 2Å) and is physically plausible.

3.4 Results

3.4.1 Docking with perfect pose classifiers

Docking Method	Redocked		Apo	
	Theirs	Perfect	Theirs	Perfect
Smina	**54.9**	86.4	11.5	36.8
Gold	53.9	70.7	**13.5**	26.4
DiffDock	27.1	40.5	8.1	19.9
Combined	-	**93.2**	-	**45.3**

Table 3.2: Performance of different docking methods (Smina, Gold, DiffDock, and a Combined approach) on the PoseBusters benchmark set, evaluated across two scenarios: redocking, docking into the cognate structures, and docking into apo, predicted by Boltz-1x, structures. Accuracies are presented using each method's native pose ranker (Theirs) and an ideal pose ranker (Perfect), and measured using the Top1 metric.

As part of the motivation for this work, I evaluated the theoretical upper limit of docking ranking accuracy using the PoseBusters benchmark set (Buttenschoen et al. 2024), a widely used standard for assessing docking methods. Throughout this chapter, I considered a pose to be accurate if it has an RMSD below 2Å and passed

all PoseBusters' physical validity checks. I selected three popular tools: Smina, a fork of AutoDock Vina optimised for pocket-specific docking; Gold, a proprietary pocket-specific method that uses genetic algorithms for conformational search; and DiffDock, a deep learning–based blind docking approach that does not require pocket definition. These tools were chosen because they explore ligand conformations by translating, rotating, and rotating rotatable bonds without modifying the protein structure or altering ligand bond lengths and atomic positions. I did not consider newer co-folding approaches, such as AlphaFold 3 (Abramson et al. 2024), and the Boltz (Wohlwend et al. 2025) and Chai (Chai-Discovery-Team et al. 2024) families of methods, but analysis could be extended to them in future work.

Typically, the field has assessed docking tools by the combined accuracy of their pose generations and whether their respective ranking method successfully ranked an accurate pose as the top pose (shown in Table 3.2). To simulate a perfect pose ranker, I selected the most accurate docked pose for each protein-ligand in the dataset and measured the Top-1 accuracy, the percentage of top-ranked poses ("Perfect"). The gap in performance in using a perfect ranker compared to the ranker of the docking software demonstrates that pose ranking is still a source of error in docking pipelines. Further, I aggregated all the output poses from the three docking methods, Smina, GOLD, and DiffDock, into a single set, referred to as "Combined". For each protein-ligand complex, I then selected an accurate pose if one existed among the combined generated poses. This approach led to improved accuracy in both redocking (docking a ligand back into its cognate receptor) and docking into a noisier apo structure, specifically one generated by Boltz-1x. These results demonstrate that the development of a universal pose classifier for any docking software could increase the accuracy of a docking pipeline. However, even with this combined approach and a perfect classifier, docking into predicted apo structures remains difficult, achieving at best 45.3% Top1 accuracy. This limitation is likely due to errors in side chain packing or interface prediction, which prevents docking algorithms from recovering the correct pose. This result demonstrates that there is still a need for classification, not just ranking of poses, to discriminate unsuccessful

docking attempts from those that did produce an accurate pose. Being able to triage whether the best generated pose is accurate or not, and with high calibration, is as crucial as getting accurate predictions for docking in structure-based drug discovery. However, it is not clear how generalisable models trained to classify poses from a single docking method will be to others, and so how "universal" they will be. By using Smina and the Redocked2020 and CrossDocked2020 datasets as a toy example, I next examined this question.

3.4.2 Adversarial impact of noise on pose probabilities

To investigate whether pose classifiers trained on Smina poses are robust to out-of-distribution poses, I scored different classes of poses with varying levels of noise that were still accurate according to the standard docking criteria (<2Å and PBValid). To generate these poses, I took crystal structures from the PoseBusters benchmark set. I corrected bond lengths and angles, and employed Latin-hypercube sampling with rejection to generate diverse poses across defined RMSD ranges. Without any further sampling, the poses were kept highly close to the original crystal pose (Fig. 3.3 ∼0 RMSD (Å)). To produce accurate poses from the biased distribution, I minimised the corrected poses under the Smina energy function (Fig. 3.3 Smina Minimised). Smina minimises by optimising pose position with a limited conformational search to improve the energetics of the pose measured by its scoring function. Further, I generated poses with RMSD within 0 and 1 Å RMSD (Fig. 3.3 $0 < \text{RMSD} < 1$ Å) and between 1 and 2Å RMSD (Fig. 3.3 $1 < \text{RMSD} < 2$ Å), which are relatively more "noisy" poses. The RMSD distributions and Smina scores of the poses are displayed in Figure 3.4. The Smina score can be used as a proxy for how out of distribution a pose is relative to what Smina would produce, and the noisier poses produce more positive and so less favourable scores. Even though the Smina Minimised poses are less accurate than the ∼0 RMSD poses, they produce slightly more favourable Smina scores.

I trained multiple versions of a simple EGNN model, based on the architecture of PointVS, named PoseTriager. I trained the first two sets of models on the

Redocked2020 set and the Crossdocked2020 subsampled set, respectively. These datasets are generated from the Smina software and, therefore, have only been exposed to poses from that distribution, and so are potentially "biased". I created a novel dataset, Augmented2020, using the same protein-ligand complexes as Redocked2020 and Crossdocked2020, but with an additional step using PoseFoundry to produce augmented poses that are "unbiased". For each dataset, I trained five versions and ensembled the predictions for this experiment. The methodology is described in further detail in the Data and Methods section of this chapter.

When trained on Crossdocked2020 and Redocked2020, the models predict higher probabilities for high accuracy poses (Fig. 3.3 ∼0 RMSD (Å)) and those from the minimisation by Smina (Fig. 3.3 Smina Minimised). When scoring these poses, Crossdocked2020 assigns higher probabilities (0.89 and 0.91, respectively) compared to the model trained on Redocked2020 (0.60 and 0.65). This is likely due to the difference in balance in the training data with actives and inactives, leading to differences in calibration. However, when scoring poses adverserially noised yet still accurate poses, both PoseTriager (Redocked2020)'s probabilities are substantially lower with medians of 0.06 and 0.02 for $0 <$ RMSD (Å) < 1 poses and $1 <$ RMSD (Å) < 2 for training on Redocked2020. For training on Crossdocked2020, the models predicted pose probability medians of 0.45 and 0.29 for $0 <$ RMSD (Å) < 1 and $1 <$ RMSD (Å) < 2, respectively. This demonstrates that small amounts of positional alterations to crystal poses, illustrated in Figure 3.5 with a case study for a single protein-ligand complex, cause a severe reduction in the confidence of a pose's quality.

Training on Crossdocked2020 is more robust and has relatively higher probabilities for these noisy poses compared to Redocked2020, which could be due to the structural noise introduced through crossdocking in the training data. A confounding factor, though, is that Crossdocked2020 generally produces higher probabilities for the in-distribution poses (∼0 RMSD (Å) and Smina Minimised), so this difference might also be a result of this behaviour. The results clearly show that the PoseTriager model trained on Augmented2020 consistently predicts very

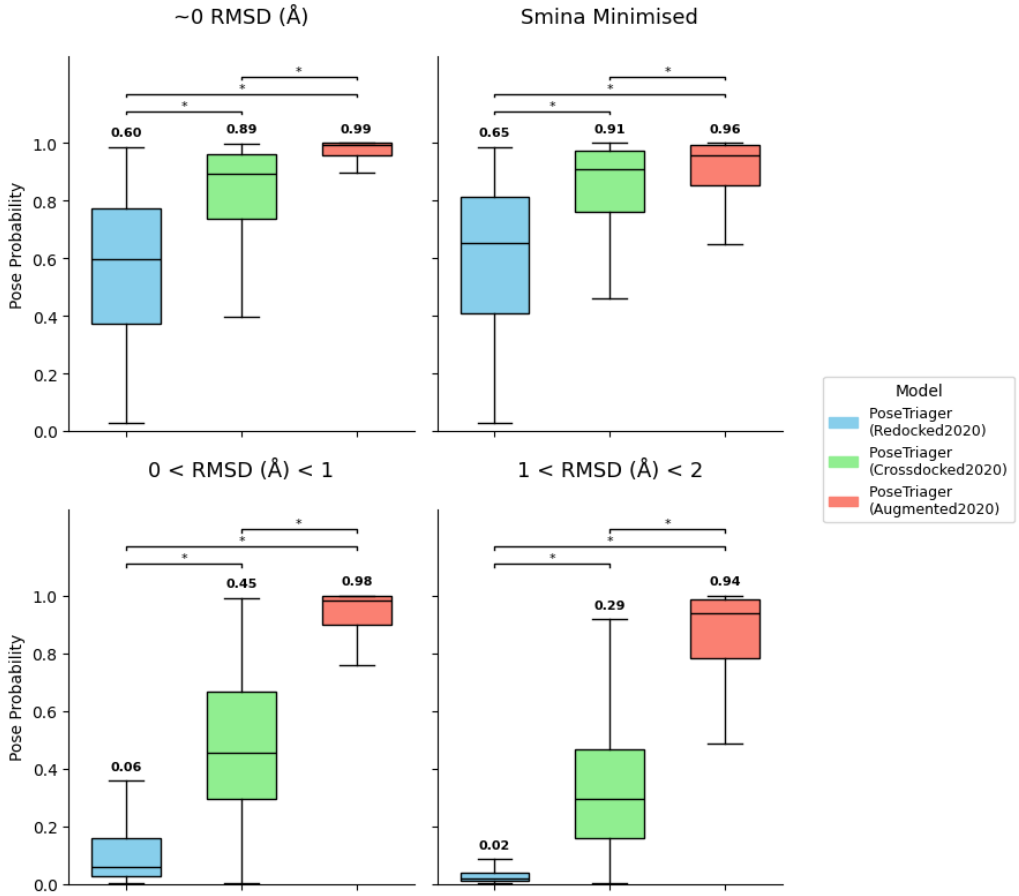
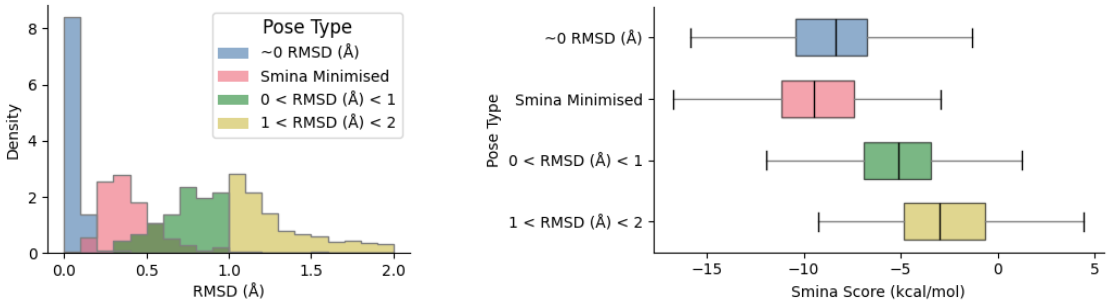


Figure 3.3: Boxplots of the distribution of different accurate (<2Å RMSD and PBValid) pose types by PoseTriager trained on the Redocked2020, Crossdocked2020 and Augmented2020 datasets. Values above the boxplot represent the median pose probability for the dataset and model combination. Significance ($p < 0.05$) in the difference between pose probability distributions is calculated using the Mann-Whitney U test and displayed as *.



(a) (b)

Figure 3.4: (A) Root mean squared deviation (RMSD) and (B) Smina scores (kcal/mol) of PoseBusters poses from ~0 RMSD (Å), Smina Minimised, 0 < RMSD (Å) < 1 and 1 < RMSD (Å) < 2.

high probabilities to accurate poses, with a median of 0.99 for ∼0 RMSD (Å), 0.96 for Smina Minimised poses, and 0.98 for 0 < RMSD (Å) < 1, and 0.94 for 1 < RMSD (Å) < 2. By training the model on this noise, the model remains capable of scoring highly on poses from distributions it was not necessarily exposed to during training, such as Smina Minimised. This suggests a general understanding of what constitutes an accurate pose that is robust to noise and potentially less biased.

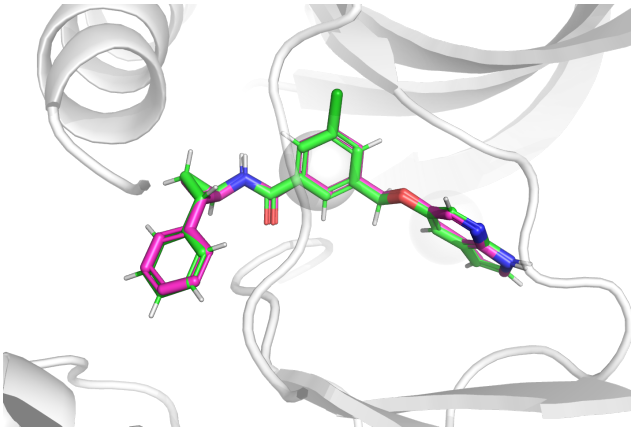
To further examine the performance of the PoseTriager model trained on the Augmented2020 model, I split the PoseBusters benchmark set into different subsets based on protein sequence similarity to the nearest member of the training set. The three ranges of protein sequence similarity were 0-30%, 30-95% and 95-100%, as shown in Figure 3.6. When training PoseTriager on CrossDocked2020's pose predictions, it predicts significantly higher probabilities for poses that are highly accurate (∼0 RMSD (Å) and Smina Minimised) and close to the training distribution (95-100%). PoseTriager (Redocked2020) shows significant differences also in these sets between 95-100% and 0-30%. When training with Augmented2020, there is no significant difference across similarity subsets except for between 95-100% and 0-30% for Smina Minimised poses. These inconsistent trends indicate that noise, rather than training data similarity, is the primary driver of the observed lack of robustness. Yet there is still some relationship between training-test similarity and pose confidence. However, it should be noted that the limited size of these subsets limits the statistical significance that can be established, so that this analysis would benefit from larger benchmark sets.

ML-based docking methods are capable of generating poses that are accurate according to metrics like RMSD; however, they are also capable of generating physically implausible poses. These predictions can either be too close to protein atoms or other ligand atoms, forming highly energetically unfavourable clashes. Physics-based methods such as AutoDock Vina or Smina are explicitly discouraged from generating these types of poses using repulsive energy terms. However, testing the ability of different trained versions of PoseTriager to detect these physical implausibilities elucidates the impact of inductive biases on model performance.

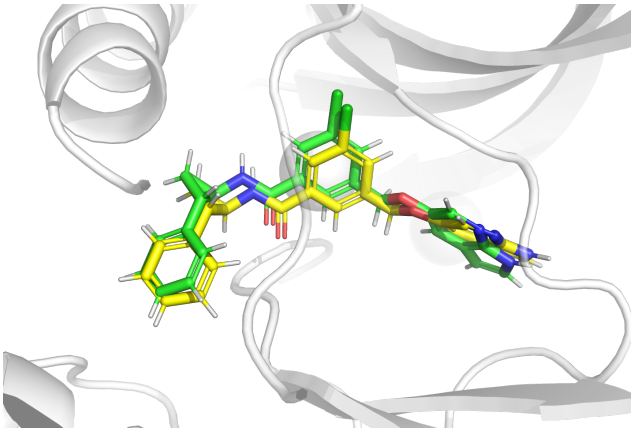
The three classes of physical invalidity are ligand-ligand clashes (Internal Clashes), a significant VdW overlap between the ligand and protein (Protein-Ligand Volume Overlap) and protein-ligand clashes (Protein-Ligand Clashes). Protein–Ligand Volume Overlap and Protein–Ligand Clashes both quantify steric clashes between the ligand and protein. Volume Overlap provides a global measure of overall clash, whereas Clashes identifies specific local conflicts. The measurement of these types of physical invalidity is described in the Data and Methods, and example structures are shown in Figure 3.7.

The Redocked2020 and Crossdocked2020 were generated using the Vina scoring function, so they contain very few examples of protein-ligand complexes with physical implausibilities. Figure 3.8 displays that all classes of physically implausible poses are scored with low probabilities when PoseTriager was trained on these datasets. By having lower confidence for these physically invalid poses outside their training distribution, they are correctly scoring them. On the other hand, Augmented2020 contains the same three examples of physically implausible poses, all within 2Å RMSD. To account for the impact of training on these poses, I additionally trained PoseTriager on the Augmented2020, but with these physically invalid poses excluded (Augmented2020 No PBInvalid). PoseTriager, trained on Augmented2020, can score these physically implausible poses negatively with a lot more consistency, with a median of 0.0. Training on Augmented2020 with no PBInvalid poses resulted in scoring these physically invalid poses with much higher median probabilities (0.44 for Internal Clashes, 0.55 for Protein-Ligand Volume Overlap and 0.66 for Protein-Ligand Clashes). Therefore, this suggests that training on Redocked and CrossDocked2020 has led the model to learn a correct inductive bias from the Smina data (that lacks these poses), and that these poses are not accurate. However, by training on increased noise and so removing this inductive bias, training on Augmented2020 requires explicit training to penalise these clear physical violations. This difference highlights the distinction between learning inductive biases and acquiring an unbiased understanding of pose accuracy. If training is done to remove

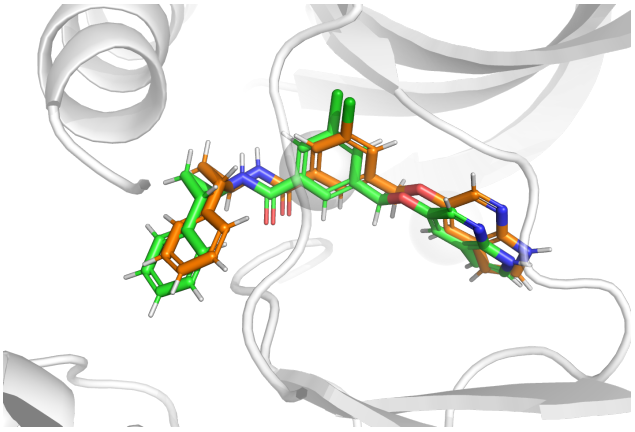
these dataset biases, which are not always accurate, the model should also be explicitly trained to relearn useful biases.



(a) Smina minimised pose (pink) of 1K2 in 5SB2; RMSD 0.18Å. Crystal pose in green.

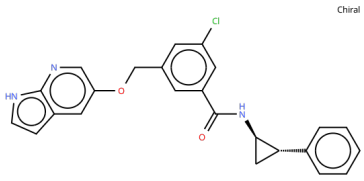


(b) Noised pose (yellow) of 1K2 in 5SB2; RMSD 0.80Å. Crystal pose in green.



(c) Noised pose (orange) of 1K2 in 5SB2; RMSD 1.02Å. Crystal pose in green.

2D Depiction of Chemical Component Directory (CCD):1K2



(a) Smina Minimised Predictions

Training set	Prob.
Redocked2020	0.93
Crossdocked2020	0.99
Augmented2020	1.00

(b) 0 < RMSD < 1 Å Predictions

Training set	Prob.
Redocked2020	0.04
Crossdocked2020	0.61
Augmented2020	1.00

(c) 1 < RMSD < 2 Å Predictions

Training set	Prob.
Redocked2020	0.02
Crossdocked2020	0.31
Augmented2020	0.99

Figure 3.5: Case study of the effect of noise on pose scoring. 3D structures (left) and PoseTriager predictions with 2D ligand depiction (right) for CCD:1K2 bound to PDB:5SB2. Model predictions for pose probabilities (Prob.) are averages over five models trained on the Redocked2020, CrossDocked2020 and Augmented2020.

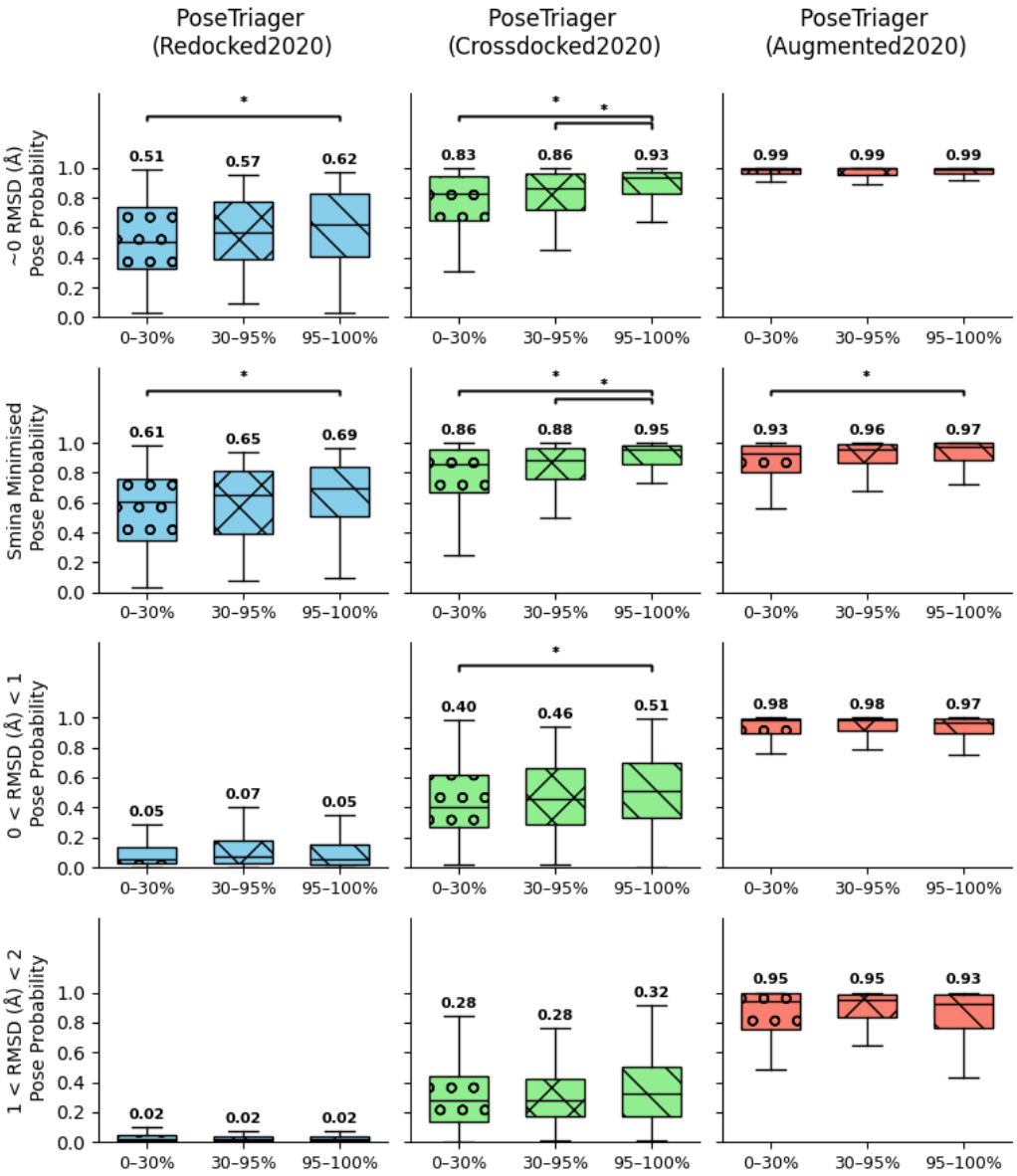
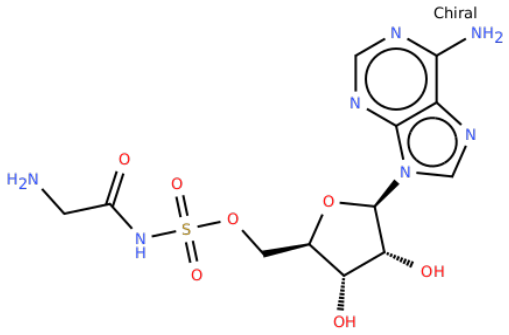
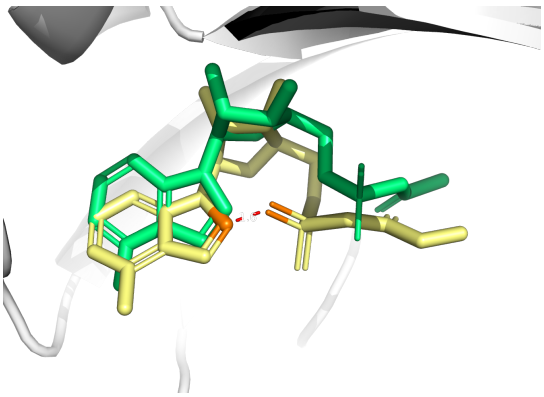


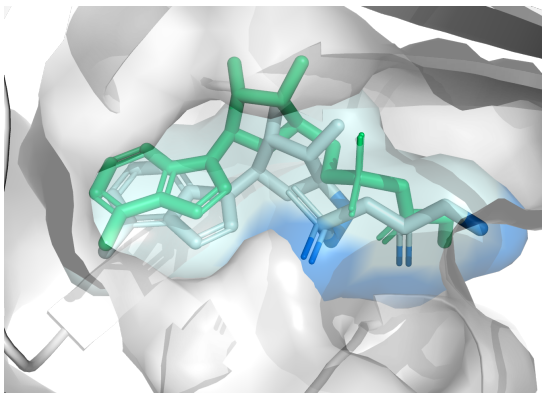
Figure 3.6: Boxplots of the distribution of different accurate (<2Å RMSD and PBValid) pose types by PoseTriager trained on the Redocked2020, Crossdocked2020 and Augmented2020 datasets for different PoseBusters benchmark subsets based on protein sequence similarity to the training set (0-30%, 30-95% and 95-100%). Median pose probabilities are displayed above the boxplots. Significance ($p < 0.05$) in the difference between pose probability distributions is calculated using the Mann-Whitney U test and displayed as *.



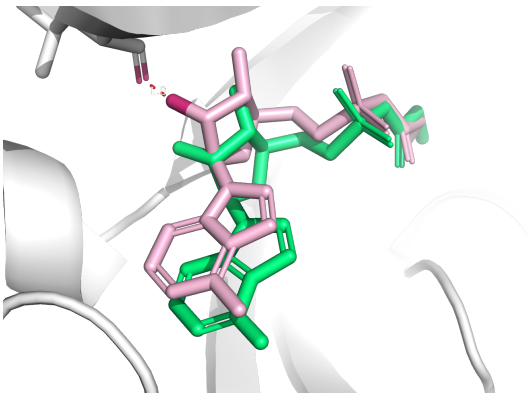
(a) 2D Chemical structure of CCD:G5A



(b) Depiction of internally clashing ligand pose for CCD:G5A bound to PDB:8SLG (yellow) within 2Å RMSD of the crystal pose. Clashing atoms are coloured orange. The crystal pose is shown in green for comparison.



(c) Depiction of ligand pose and its VdW volume overlapping with protein (Protein-Ligand Volume Overlap) for CCD:G5A bound to PDB:8SLG (blue) within 2Å RMSD of the crystal pose. Volume overlap is coloured in dark blue. The crystal pose is shown in green for comparison.



(d) Depiction of ligand pose clashing with protein atom (Protein-Ligand Clashes) for CCD:G5A bound to PDB:8SLG (pink) within 2Å RMSD of the crystal pose. Clashing atoms are depicted in red. The crystal pose is shown in green for comparison.

Figure 3.7: Specific examples of PoseBusters failures of CCD:G5A (a) bound to PDB:8SLG: (b) Internal Clash, (c) Protein-Ligand Volume Overlap and (d) Protein-Ligand Clashes.

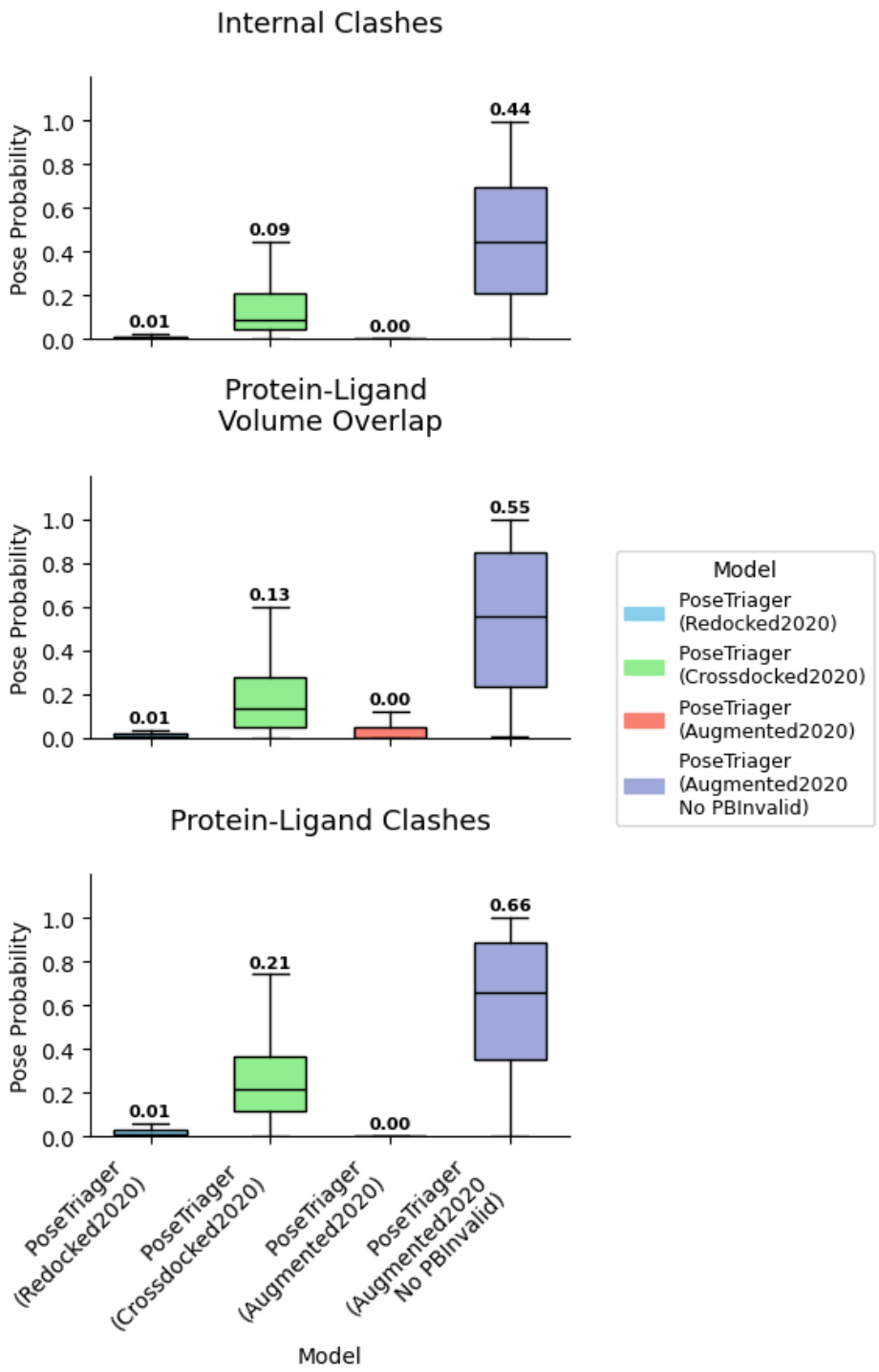


Figure 3.8: Boxplots of pose probabilities from different PoseTriager models trained on the Redocked2020, Crossdocked2020, Augmented2020 and Augmented2020-No PBInvalid datasets for poses that fail specific PoseBusters physical validity checks (Internal Clashes, Protein-Ligand Volume Overlap and Protein-Ligand Clashes). Median pose probabilities are displayed above the boxplots.

3.4.3 Redocked pose classification accuracy

Having established that training on Smina generates models that are not robust to artificially noisy poses outside of their training distributions, I explored whether improving this robustness helps generalisation. Specifically, I tested generalisation for pose classification on poses generated by other docking software that the models are not trained on. To do this, I took the trained PoseTriager models on different datasets and scored the docked poses of the PoseBusters benchmark of Smina (Koes et al. 2013), Gold (Verdonk et al. 2003) and DiffDock (Corso et al. 2022). Smina represents an in-distribution pose classification test, as all models are trained on poses originating from this software. Gold represents a similar method that also samples poses based on a related, but not identical, physics-based function. Therefore, although not directly in-distribution, the poses outputted by this software would be expected to be similar to those of Smina. DiffDock, however, samples poses by denoising the torsional-translation manifold of a ligand and so does not produce poses according to a known physics function and so is more likely to make similar poses to the out-of-distribution poses generated by PoseFoundry. I measured pose classification using the MCC and ranking accuracy using the AUROC, AUPRC and the Top-1 accuracy. The thresholds for classification were optimised based on the highest MCC possible on each model's validation set and are displayed below (Table 3.3). First, I examined pose classification accuracy on poses from redocking the poses into the cognate structure with the three docking methods.

Table 3.3: Classification thresholds for each model for repeat (1-5) and training dataset (Redocked2020, Crossdocked2020 and Augmented2020).

Repeat	Redocked2020	Crossdocked2020	Augmented2020
1	0.25	0.34	0.50
2	0.29	0.39	0.40
3	0.15	0.44	0.37
4	0.42	0.32	0.59
5	0.40	0.44	0.48

For the in-distribution Smina Redocking benchmark, training on augmented poses (Augmented2020) harms accuracy compared to training on Smina-generated

Table 3.4: Performance of PoseTriager models trained on Redocked2020, Crossdocked2020 and Augmented2020 on the Smina redocking benchmark from PoseBusters. Classification metric values are mean ± 95% confidence intervals across five trained models. Bold indicates the best performance per metric.

Metric	Redocked2020	Crossdocked2020	Augmented2020
MCC	**0.44±0.02**	0.34±0.02	0.34±0.04
AUROC	**0.90±0.01**	**0.90±0.01**	0.85 ± 0.03
AUPRC	**0.45±0.02**	**0.47± 0.02**	0.37 ± 0.05
Top-1 Accuracy	**0.59±0.03**	0.55±0.02	**0.58±0.01**

poses of Redocked2020 and Crossdocked2020 in all metrics except Top-1 Accuracy, as shown in Table 3.4. Training on Redocked2020 provides the highest accuracy in all metrics, as there is the least distributional change between the training and test set. The PoseTriager models trained on Redocked2020 and Crossdocked2020 are likely to have learnt useful inductive biases to rank poses appropriately. Removing these biases by training on increased noise is harming accuracy.

Table 3.5: Performance of PoseTriager models trained on Redocked2020, Crossdocked2020 and Augmented2020 on the Gold redocking benchmark from PoseBusters. Classification metric values are mean ± 95% confidence intervals across five trained models. Bold indicates the best performance per metric.

Metric	Redocked2020	Crossdocked2020	Augmented2020
MCC	**0.60±0.03**	**0.62±0.02**	0.56±0.04
AUROC	**0.92±0.01**	0.90± 0.01	0.86 ± 0.02
AUPRC	**0.89±0.01**	0.86± 0.01	0.82 ± 0.02
Top-1 Accuracy	**0.59±0.01**	**0.59±0.01**	**0.59±0.01**

For the Gold Redocking benchmark (Table 3.5), the trends are similar, with training on Augmented2020 still having reduced accuracy compared to Redocked2020 and Crossdocked2020 trained models. This result supports my hypothesis that the poses generated by Gold and Smina exhibit similar distributions, indicating minimal distribution shift. Notably, all methods perform identically on Top-1 accuracy (0.59).

When docking using DiffDock into the cognate structure, the performance of training on Augmented2020 is within confidence intervals of training on Crossdocked2020 in all metrics, as shown in Table 3.6. Training on Redocked2020, however, results in similar ranking accuracy, measured by AUPRC and AUROC, yet has a lower MCC (0.29). However, despite this, PoseTriager (Redocked2020) can still

Table 3.6: Performance of PoseTriager models trained on Redocked2020, CrossDocked2020 and Augmented2020 on the DiffDock redocking benchmark from PoseBusters. Classification metric values are mean ± 95% confidence intervals across five trained models. Bold indicates the best performance per metric.

Metric	Redocked2020	Crossdocked2020	Augmented2020
MCC	0.29±0.02	**0.41±0.03**	**0.40±0.06**
AUROC	0.79±0.01	**0.81± 0.01**	**0.79±0.02**
AUPRC	**0.92±0.01**	**0.92± 0.01**	**0.91±0.01**
Top-1 Accuracy	**0.39±0.00**	**0.38±0.01**	**0.38±0.01**

rank and pick out positive classes, based on the ranking alone, with similar accuracy to the other models. This suggests that the distribution shift of the poses does not affect the model's ability to rank, but it does affect the absolute pose probabilities given to these poses. However, testing the ability of methods when redocking is an unrealistic docking scenario, as it requires the cognate receptor to already exist, negating the need for docking. To address this limitation, I explored the accuracy of the different models' classification when docking into predicted apo structures.

3.4.4 Apo-docked pose classification accuracy

Table 3.7: Performance of PoseTriager models trained on Redocked2020, CrossDocked2020 and Augmented2020 on the Smina apo benchmark from PoseBusters. Classification metric values are mean ± 95% confidence intervals across five trained models. Bold indicates the best performance per metric.

Metric	Redocked2020	Crossdocked2020	Augmented2020
MCC	0.21±0.02	**0.24±0.02**	**0.22±0.03**
AUROC	0.87±0.01	**0.90± 0.01**	0.82±0.03
AUPRC	0.16±0.01	**0.20± 0.02**	0.13±0.04
Top-1 Accuracy	0.12±0.01	**0.15±0.01**	**0.16±0.01**

To reflect a noisier docking scenario, I docked the ligands of the PoseBusters benchmark into predicted co-folded structures by Boltz-1x. These cofolded structures do not have the bound ligand, but other cofactors are included in the prediction. Cofolding generates noisier, inaccurate conformations, making it harder to score predicted poses, as the interactions and energy of the pose will not be optimal. First, I docked into these apo structures using Smina and scored as done in the previous section. Table 3.7 shows that training on the increased noise of Augmented2020

does not outperform training on CrossDocked2020 scoring accuracy with all metrics except Top-1 Accuracy (0.16), performing worse. Redocked2020 performs worse in other metrics compared to Crossdocked2020, indicating that training on cross-docked structures is a better strategy for accounting for this structural noise when docking with Smina, compared to the approach I have proposed, PoseFoundry.

Table 3.8: Performance of PoseTriager models trained on Redocked2020, CrossDocked2020 and Augmented2020 on the Gold apo benchmark from PoseBusters. Classification metric values are mean ± 95% confidence intervals across five trained models. Bold indicates the best performance per metric.

Metric	Redocked2020	Crossdocked2020	Augmented2020
MCC	0.36±0.04	**0.45±0.01**	0.41±0.03
AUROC	**0.87±0.01**	**0.87± 0.01**	0.83±0.02
AUPRC	**0.53±0.03**	**0.53± 0.00**	0.49±0.01
Top-1 Accuracy	**0.16±0.01**	**0.16±0.01**	**0.17±0.00**

Further, I also examined the pose classification accuracy using Gold docking into apo structures and found a similar trend (Table 3.8). Training on the increased noise of Augmented2020 did not help exceed the performance of training on Crossdocked2020. Again, the Top1 accuracy was within confidence intervals for all model types. For the final benchmark, I docked into apo structures using

Table 3.9: Performance of PoseTriager models trained on Redocked2020, CrossDocked2020 and Augmented2020 on the DiffDock apo benchmark from PoseBusters. Classification metric values are mean ± 95% confidence intervals across five trained models. Bold indicates the best performance per metric.

Metric	Redocked2020	Crossdocked2020	Augmented2020
MCC	0.22±0.04	0.17±0.05	**0.27±0.03**
AUROC	0.63±0.03	0.57± 0.04	**0.67±0.02**
AUPRC	0.61±0.02	0.57± 0.03	**0.64±0.02**
Top-1 Accuracy	**0.17±0.01**	0.16±0.00	**0.17±0.00**

DiffDock, which presents a highly out-of-distribution test compared to redocking with Smina. Here, training on the Augmented2020 does result in a significant increase in accuracy on all metrics except Top-1 accuracy, as shown in Table 3.9 compared to Redocked2020 and Crossdocked2020. Overall, this demonstrates that training on Augmented2020 and accounting for this lack of robustness only improves

pose classification accuracy when the poses are highly noisy. For scoring Smina and Gold-generated poses, where docking perturbations are smaller and closer to the training distributions, Crossdocked2020 remains superior. Crossdocked2020 appears optimal for conventional docking into apo/holo structures with physics-based methods, while Augmented2020 may be beneficial when the target inference scenario involves large structural deviations or ML-based docking methods such as DiffDock. Therefore, these results suggest that improving generalisability across docking methods is not going to be primarily driven by improving robustness to noise in training. Pose classifier dataset choice should be guided by the expected noise characteristics of the intended docking pipeline.

3.5 Discussion

This work aimed to explore the impact of learning pose quality from a "biased" distribution of poses generated by a single docking method and how that affects generalisation to pose classification for other methods. I have demonstrated the value of developing a universal pose classifier by exploring the upper limit of Top-1 accuracy for docking methods and how collating poses from these different methods could result in an overall more accurate docking methodology. Developing such a methodology would require training pose classifiers that were able to generalise to the different poses generated by various types of software. I used a simple EGNN to create the PoseTriager model and the Smina-generated Redocked2020 and Crossdocked2020 datasets to explore the impact of training a pose classifier on a single source of poses on generalisability. By simply training on this higher noise, using my Augmented2020 dataset, the models were more robust to this noise and retained high confidence in adversarially noised poses. Models trained on Redocked2020 and CrossDocked2020 were not robust to this noise. However, by training on noise and removing this inductive bias, the model needs to be explicitly trained to recognise obvious inadequacies in pose quality, which can typically be measured using distance-based thresholds.

Having established this lack of robustness and addressed it by creating the

Augmented2020 dataset, I tested whether this new robustness to noise helped improve generalisability for pose classification to other docking software. Models trained on Crossdocked2020 generalise best to conventional physics-based docking, while Redocked2020 is optimal when testing on poses close to its training distribution, such as redocking with physics-based methods like Smina and Gold. Augmented2020, despite its lack of improvement on most benchmarks, promises substantial gains for highly out-of-distribution cases, such as ML-based docking (in my case, DiffDock), into predicted apo structures, where the noise magnitude and diversity match its training set. However, these highly noisy structures, although accurate according to the RMSD error thresholds, may not be meaningfully useful as the original interactions between protein and ligand may not be recapitulated (Errington et al. 2025).

While models trained on cleaner datasets lack robustness to high-noise poses, this limitation does not necessarily impair their performance on poses from other docking software. Improving this robustness might have further applications, though, in prioritising which *de novo* generated compounds, which are often structurally noisy (Harris et al. 2023), to test experimentally or as a guidance function for generative models. One major limitation of this study is that it only considers pose quality through the RMSD metric. It has been shown that interaction similarity is a more precise metric (Errington et al. 2025). However, it is not clear what interactions are key for binding and which are merely artefacts of the experimental structure model-building process. PoseFoundry could be applied to generate training data for a classifier that discriminates based on more restrictive, important interaction similarity criteria. While this chapter relied on poses derived from experimentally determined protein–ligand complexes in the PDB, the following chapter investigates the generation of synthetic complex data through ligand pocket design. Such different augmentation of structural datasets has the potential to enhance the accuracy and generalisability of pose classifiers.

4

On the potential of ligand pocket design to synthetically expand the structural pocketome

Contents

4. On the potential of ligand pocket design to synthetically expand the structural pocketome

4.1 Preface

The field of machine learning-based structure-based design for small molecule drugs is currently limited to the structures deposited to the PDB. For example, the available protein-ligand structural data from PLINDER (Durairaj et al. 2024), the largest public dataset curated from the PDB, is limited in number (113,564) and in diversity, with 50% of protein-ligand complex data available in the data coming from just 4.9% of unique proteins (uniqueness measured by UniProtID) and 2.4% of unique ligands (uniqueness measured by CCD ID) (shown in Figure 1.11). Tackling this problem could drive improvements in the field. In this chapter, I examine methods to address this data paucity using ligand pocket design. Ligand pocket design is the sequence prediction of an existing structural scaffold to bind a specified ligand. The side chain orientation, backbone, and ligand position are sometimes also predicted alongside the pocket amino acid sequence. Normally, this methodology is used for the design of novel catalysts (Jiang et al. 2008; Röthlisberger et al. 2008), biosensors (Beltrán et al. 2022; Bick et al. 2017), and signal transducers in cells (Santos et al. 2016; Chen et al. 2020). However, in this work, I aim to apply these methods to enumerate under-represented proteins or ligands in the PDB (Burley et al. 2017) to both increase the diversity of data and to increase its total amount.

Expanding the available structural data has been attempted previously: through methods such as breaking up the protein and treating fragments as ligands (Krishna et al. 2024; Corso et al. 2023), creating synthetic interactions around ligands to over-sample known interactions (Voitsitskyi et al. 2024), with no realistic protein and docking known binders (Zhu et al. 2025). These methods are limited in that there is a substantial distributional shift from the experimentally determined protein-ligand complexes normally used for training to these noisy, physically unrealistic structures. In this chapter, I test whether ML methodologies for ligand pocket design (such as LigandMPNN (Dauparas et al. 2023), PocketGen (Zhang et al. 2024b) and FlowSite (Stark et al. 2023)) can generate plausible novel structures for training and explore whether these different methods, ranging in complexity and architectures, are good enough to generate useful synthetic structural data for training other ML methods.

4. On the potential of ligand pocket design to synthetically expand the structural pocketome

I found that all the methods tested were not able to generate useful data. This work, however, enables others to benchmark other potential methods.

4.2 Introduction

De novo protein design using ML and AI techniques has become an exciting possibility in recent years, replacing and augmenting previous slower and more limited computational methods (Ding et al. 2022; Notin et al. 2024). Advances in generative AI in other fields, such as natural language processing (Achiam et al. 2023; Dubey et al. 2024; Gozalo-Brizuela et al. 2023) and image generation using large language models and diffusion models (Ho et al. 2022b; Batzolis et al. 2021), have been translated into protein design (Watson et al. 2023; Dauparas et al. 2022). Methods for generating proteins include sequence-based methods (Lin et al. 2023; Brandes et al. 2022; Olsen et al. 2022b), often using large language models trained on large corpora of protein sequence data (Olsen et al. 2022a; Boutet et al. 2007; Mitchell et al. 2020). The large volume of data enables these models to learn expressive representations of protein sequences, which are useful for downstream prediction tasks (Lin et al. 2023; Brandes et al. 2022; Schmirler et al. 2024). It is also possible to generate backbone structure and thus the 3D structure of the protein directly (Watson et al. 2023; Yim et al. 2023b; Yim et al. 2023a; Ingraham et al. 2023). The backbones can also be conditioned on other moieties such as DNA and small molecules by methods like RFDiffusionAA (Krishna et al. 2024). The amino acid sequences of these generated, or even experimentally determined, backbones can be predicted with the same model or by a separate model. This separate backbone-conditioned sequence design is often referred to as inverse folding (Dauparas et al. 2022; Gao et al. 2022; Hsu et al. 2022). An emerging application of inverse folding is the design of pockets that bind specific ligands using these structure-conditioned sequence designers. The design of protein pockets has a broad range of applications, including the design of novel catalysts (Jiang et al. 2008; Röthlisberger et al. 2008), biosensors (Beltrán et al. 2022; Bick et al. 2017), and signal transducers in cells (Santos et al. 2016; Chen et al. 2020).

4. On the potential of ligand pocket design to synthetically expand the structural pocketome

Ligand pocket design typically involves the prediction of the amino acid identities of the residues interacting with the ligand in the pocket. The orientation of the side chains can also be predicted; they provide atomistic detail of how the proposed sequence might interact with the ligand. Previously, the design of pockets has been achieved using physics-based methods such as Rosetta (Leman et al. 2020) and PocketOptimiser (Malisi et al. 2012; Noske et al. 2023), which both employ libraries of rotamers for side chains and optimise both the amino acid identity and the rotamers of the side chains given a ligand conformer. Polizzi et al. developed a template-based method by fragmenting the pocket into individual units and assigning them based on frequencies in proximity to parts of the ligand from their structural database. However, these methodologies are computationally expensive and unsuited to the high-throughput generation of structures. Researchers have proposed deep learning methods promising faster generation and more reliable predictions. These methods co-predict an amino acid sequence and a structure for the pocket, conditioned on the ligand. They are primarily trained on existing structures of protein-ligand complexes to accurately re-predict the original sequence of the pocket. Different methods have different capabilities as to which parts of the structure they co-predict (see Table 4.1), but ML's flexibility enables both the ligand pose and backbone position to be predicted as part of this co-prediction.

Capability	**LigandMPNN**	**PocketGen**	**FlowSite**
Pocket residue identity	✓	✓	✓
Side-chain orientation	✓	✓	✓
Ligand position	✗	Updates only	✓
Backbone position	✗	Updates only	✗
Details	Random decoding of residue identities based on local environment	Transformer adapting ESM2 embeddings with ligand pose	Flow matching for docking with iterative residue prediction

Table 4.1: Comparison of ligand pocket design models by structural prediction capability.

4. On the potential of ligand pocket design to synthetically expand the structural pocketome

LigandMPNN (Dauparas et al. 2023), based on the structure-conditioned sequence model ProteinMPNN (Dauparas et al. 2022), is an encoder-decoder GNN model that sequentially predicts amino acid identities in a random order. The model was trained on 19,700 high-quality single-chain structures, bound with small molecules, nucleic acids and metal ions, extracted from the PDB. To allow the encoding of the atomic structure of the ligand, a further protein-ligand encoder is used to combine the protein backbone encoder with the protein-ligand graph. However, the model does not update the pose during this decoding and predicting side chain conformations requires a separate version of the model that predicts based on the amino acid side chain prediction (I refer to this model as LigandMPNN Side Chain Packer (LigandMPNNSCPacker)). The sequential prediction of amino acid identity and rotamers may introduce inaccuracies, as identity predictions can be influenced by the orientation and proximity of surrounding residues.

Another method, PocketGen (Zhang et al. 2024b), utilises a transformer architecture and a structure adaptor to incorporate the embeddings from the ESM2 protein language model (Lin et al. 2023) into its prediction of the amino acid identities conditioned on the ligand graph. The model directly predicts the atomic positions of 14 atoms for all side chains and prunes redundant atoms once it has completed its iterative predictions. PocketGen, therefore, can predict concurrently the amino acid identity and the 3D structure of those amino acid side chains. It also updates the ligand and backbone positions throughout the decoding process, but it still requires accurate, pre-defined positions that it modifies. The method also includes a post-processing step to relax the protein pocket structure using a harmonic potential and the AMBER ff99SB forcefield (Hornak et al. 2006). Finally, FlowSite (Stark et al. 2023) is one of the first models to design the pocket and predict the pose position simultaneously, even able to do so for multiple ligands at once. The method utilises flow matching (Lipman et al. 2022) over the atomic Cartesian coordinates of the ligand and iteratively predicts amino acid identity based on the intermediate pose position. The authors also trained FlowSite to predict chi angles for each amino acid side-chain as an auxiliary task.

4. On the potential of ligand pocket design to synthetically expand the structural pocketome

Given these advances, I tested that the generation of synthetic ligand-protein complexes using such methods could offer an additional data source to train structure-based deep learning methods. To be able to train structure-based models for pose prediction, binding affinity prediction and de novo molecule generation on the scale of data that exists within other deep learning domains, such as AlphaFold 2 (Jumper et al. 2021), ChatGPT and DALLE-3, significantly more and more diverse structural data is needed.

One method for increasing the amount of protein-ligand data available, employed in RFDiffusionAA (Krishna et al. 2024) and DiffDock-L (Corso et al. 2023), was to fragment sections of the protein and treat these polypeptides or molecules as a ligand binding to the cavity left behind. This "atomisation" or "van der Mer" training data improved accuracy, but did not significantly increase the amount of training data. This additional training data is further limited in that it does not consist of the drug-like compound-protein interactions that are of interest for training these methods. Another proposed methodology, synthetically generated pockets for ligands, was used to expand the training structural data and so improve the accuracy of docking (Voitsitskyi et al. 2024). To generate synthetic pockets, amino acids and dipeptides were sampled to form interactions that exist at the same rate as found in PDBBind. However, these pockets were not true proteins but rather clouds of residues. This additional training data did result in limited improvements in the ability of a specific diffusion docking software, with training on synthetic pockets outperforming the model trained on experimental pockets on some metrics, but this success can be built upon.

In this chapter, I investigate expanding the available structural data for training structure-based ML methods by upsampling the underrepresented protein-ligand complexes (displayed in Figure 1.11) by redesigning protein pockets that increase or maintain the binding of the crystallised ligand.

As shown in previous chapters, deep learning-based method development requires robust validation to drive improvement, but currently, the ligand pocket generation methods described above have not been tested rigorously. This chapter outlines the

development of a testing framework to understand whether these ligand-conditioned pocket generation methods are able to make useful predictions or are learning unhelpful patterns in the data. In doing so, I also evaluated the capabilities of these methods to ascertain which can generate useful synthetic structural data for training other ML methods.

First, I applied and extended the PoseBusters plausibility checks (Buttenschoen et al. 2024) to include the plausibility of side chains and their side chain-side chains interactions, and demonstrated that these methods struggle to produce physically plausible ligand-pocket structures as part of their co-prediction. Furthermore, I demonstrated that this implausibility confounds downstream analysis of these structures with a baseline method that exploits this implausibility to produce Vina scores (Trott et al. 2010), a common metric used for analysing binding between generated protein-ligand complexes. Second, I examined each method's ability to recapitulate the original protein pocket, measured by amino acid recovery, sequence for crystal structures and their ability to explore other potentially binding sequences. To do this, experimental deep mutational scanning data for the MET kinase were used to explore the capabilities of methods to predict binding-improving mutations for differing inhibitors. This test indicated that the models are unable to generate sequences beyond just recapitulation of the original sequence, suggesting that they have not learnt to interpolate from the data they were trained on. Finally, I also proposed adversarial tests to test the responsiveness of the methods to small changes in the ligand, showing that PocketGen is not responsive to these changes at all, whilst LigandMPNN is.

4.3 Data and Methods

4.3.1 Benchmark Data

Crystal Structures

To compare the performance of ligand-conditioned protein pocket generation methods, I used the approach of Buttenschoen et al. 2024 to use an in-distribution set and an out-of-distribution (OOD) set. For the in-distribution, I also used

the Astex Diverse Set, 85 diverse protein-ligand complexes from protein sequence clusters that were of direct interest for the pharmaceutical industry. Therefore, the dataset consists of highly represented proteins in the PDB. For the OOD set, I chose the "Runs N' Pose" set, curated originally to test the generalisability of co-folding methods (Škrinjar et al. 2025). The original dataset consists of all complexes from the PLINDER dataset after 30th September 2021 (2585 complexes). These complexes were clustered by pocket SuCOS score, a shape and pharmacophoric colour similarity metric, similarity (Leung et al. 2019), and these clusters were labelled with maximum SuCOS pocket similarity to any PDB before the cutoff. These datasets are described further in Section 1.5.2. I retained clusters that were below 50% similarity and further removed PDBs that had ligand-interacting symmetry mates, measured using PyMOL functionality (Schrödinger, LLC 2015). Finally, I removed any clusters with members deposited before 16th December 2022 (LigandMPNN's training data time cutoff). The number of protein-ligand complexes remaining after each step is depicted in Table (4.2). The final cleaned version of "Runs N' Poses" for benchmarking was 299 protein-ligand complexes. I refer to this filtered version of the benchmark as "Run N Poses" for brevity. The protein PDB and SDF files for protein and ligands were downloaded from the PDB using its API (PDB 2023), and waters were removed from the structures. PDB IDs of the protein and CCD IDs of the ligand are provided in the Appendix (C.1).

Table 4.2: Filtering steps applied to the "Runs N' Poses" OOD benchmark dataset.

Filtering Step	**No. of Remaining Complexes**
Initial Runs N' Poses complexes (after 30 Sept 2021 cutoff)	2061
SuCOS pocket similarity $< 50\%$	580
Remove ligand-interacting symmetry mates	471
Date $>$ 16 Dec 2022	299

Deep Mutational Scanning of the MET Kinase

Deep mutational scanning data for the MET kinase was sourced from Estevam et al., (Estevam et al. 2024). This study did a deep mutational scan of 5764 MET

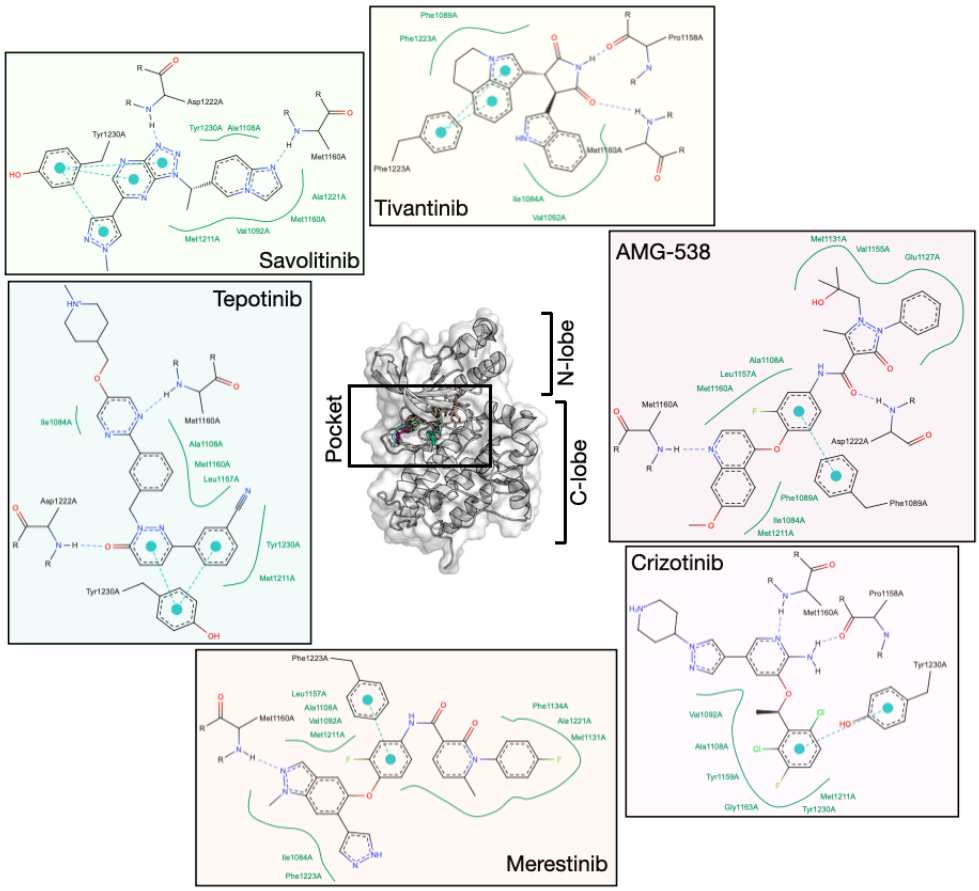


Figure 4.1: The MET kinase (inset, PDB:3DKC) depicting the N-lobe, C-lobe with pocket bound by inhibitors. The inhibitors shown are crizotinib (pink), tepotinib (aquamarine), savolitinib (green), merestinib (orange), AMG-458 (purple) and tivantinib (yellow). For each inhibitor, 2D interaction diagrams by PoseEdit (Diedrich et al. 2023) are presented, displaying hydrogen bonds (blue dashes), pi-stacking interactions (teal dashes) and hydrophobic sub-pockets (green full lines).

kinase variants, profiling the growth of Ba/F3 cells harbouring a specific mutation for 11 inhibitors. Of these 11 inhibitors, 6 had existing crystal structures of the protein-inhibitor complex, so I analysed model performance only for those six. This was to avoid the additional noise introduced by docking the remaining inhibitors. The six inhibitors were crizotinib (PDB: 2WGJ), tepotinib (PDB: 4R1V), savolitinib (PDB: 6SDE), merestinib (PDB: 4EEV), AMG-458 (PDB: 5T3Q) and tivantinib (PDB: 3RHK). These compounds with their interactions visualised and the overall structure of the MET Kinase are depicted in Figure 4.1.

To classify mutations, I fit three normal distributions to the normalised mutation scores for both the dimethyl sulfoxide (DMSO) control and each inhibitor condition.

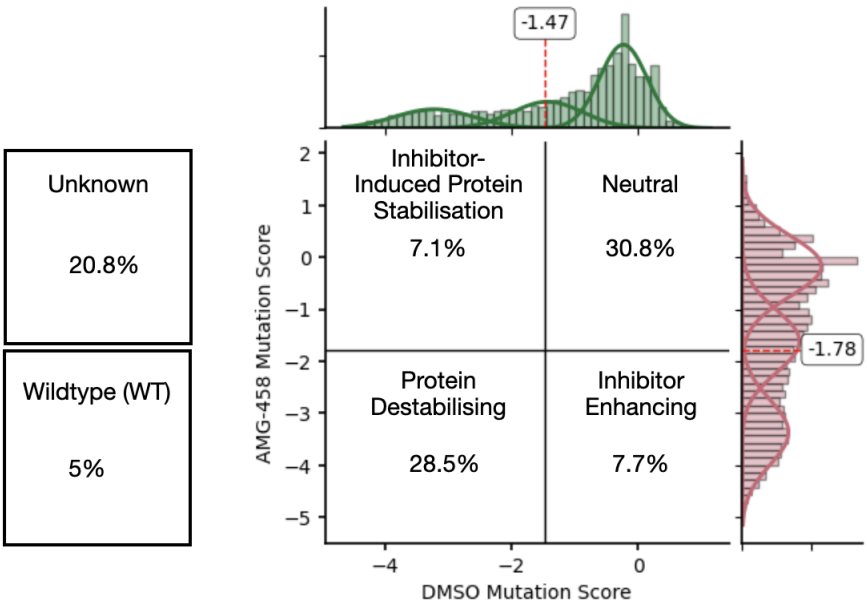


Figure 4.2: Plot and classification of mutations based on sequence reads from deep mutational scan data of the MET Kinase when assayed with a DMSO control and the inhibitor AMG-458. The normalised sequence reads ("Mutation Score") are plotted with the three Gaussian distributions fitted to them. The intersection of the outer Gaussians is the threshold used for classification.

Thresholds were defined by the intersection of the outer distributions, following Cagiada et al. (Cagiada et al. 2021) (see Fig. 4.2). Mutations with scores above the threshold under both DMSO and inhibitor were considered neutral. If the score fell below the threshold only in the presence of an inhibitor, the mutation was classified as inhibitor-enhancing, suggesting improved inhibitor binding due to reduced kinase activity. Mutations with scores below the threshold under both conditions were classified as protein destabilising. Finally, if a mutation scored below the threshold under DMSO but rose above it with the inhibitor, it was considered inhibitor-induced protein stabilising, implying that inhibitor binding rescued an otherwise inactive variant.

4.3.2 Models and Baselines

For predictions, the pocket was defined as any residue with at least one atom within 5Å of any ligand atom. LigandMPNN was run with residues beyond the 5Å threshold fixed. PocketGen was run with a 5Å threshold with its final step of minimisation of

the pocket either included (PocketGen (w/ relax)) or not (PocketGen (no relax)) to account for the impact of the external processing step on the method, which other methods do not include. FlowSite's chi angle predictions for side chains were used to predict structures using the OpenFold codebase (Ahdritz et al. 2024). Furthermore, FlowSite does not allow a pocket to be defined below a threshold of 8Å, so I used this higher threshold but disregarded redesigned residues outside of the 5Å threshold for analysis of the results.

I also tested a simple baseline that would pack the largest side chains, named VolumeBaseline. I iterated over amino acids ordered by their volume, assigned the pocket residues the amino acid identity and predicted their atomic coordinates using LigandMPNNSCPacker. If any side chain atom with Van der Waal (VdW) radius (R_1) clashes with any ligand atom, with VdW radius (R_2), its orientation was re-predicted with the next largest amino acid by volume, as defined by this simple inequality:

$$d + 0.5 < R_1 + R_2 \tag{4.3.2.1}$$

A pseudo-algorithm outlining the steps of the baseline is given in Algorithm 1.

4.3.3 Adverserial ligand change tests

To test the impact of local ligand changes on residue identity predictions, I modified the ligand at specific atomic locations using RDKit. First, I took the predictions from each model and the Astex or Runs N Poses sets and predicted the protein-ligand interactions using PLIP (Salentin et al. 2015). For each hydrogen bond or hydrophobic interaction, I altered the atom type of the ligand forming that interaction so that the interaction could no longer occur. I excluded multiple interactions formed with the same amino acid from this analysis, and those formed with the backbone for simplicity. For the "Swap" version of the adversarial tests, all carbons forming interactions were changed or "swapped" to nitrogen. Any oxygen and nitrogen atoms forming interactions were "swapped" into carbon. To

Algorithm 1 VolumeBaseline Pseudo Algorithm

Input:
Set of amino acids $A = \{a_1, a_2, \ldots, a_n\}$
Set of corresponding volumes $V = \{v_1, v_2, \ldots, v_n\}$ where $v_i \in \mathbb{R}^+$
Set of pocket residues $R = \{r_1, r_2, \ldots, r_m\}$
Set of ligand atom positions $L = \{l_1, l_2, \ldots, l_k\}$ where $l_i \in \mathbb{R}^3$
Output:
Function $f : R \to A \cup \{\emptyset\}$ mapping pocket residues to assigned amino acids or empty set if unassigned
Define a bijective function $\sigma : \{1, 2, \ldots, n\} \to \{1, 2, \ldots, n\}$ such that:
$\forall i, j \in \{1, 2, \ldots, n-1\} : i < j \Rightarrow v_{\sigma(i)} \geq v_{\sigma(j)}$
Initialize $f(r) = \emptyset$ for all $r \in R$
for $i = 1$ to n **do**
    Let $a = a_{\sigma(i)}$
    for each $r_j \in R$ where $f(r_j) = \emptyset$ **do**
        Let $X = \text{LigandMPNNSCP}(a, r_j) = \{x_1, x_2, \ldots, x_p\}$ where $x_i \in \mathbb{R}^3$
        Define the clash indicator function:
        $$C(x, y) = \begin{cases} 1 & \text{if } ||x - y|| < (R_x + R_y - 10.5) \\ 0 & \text{otherwise} \end{cases}$$
        where $R_x, R_y \in \mathbb{R}^+$ are the Van der Waals radii of atoms x and y respectively
        Compute the total clash indicator:
        $T = \sum_{x \in X} \sum_{y \in L} C(x, y)$
        if $T = 0$ **then**
            Set $f(r_j) = a$
        end if
    end for
end for
return the function f

ensure the chemical validity of the new molecules, I processed the molecules using the MoleculeRectifier package (Ferla 2021). The mutated ligands replaced the original ligand input, and then the model predicted the ligand-pocket complex again, with the seed kept constant. A change in the predicted residue indicated local sensitivity to the ligand perturbation; unchanged predictions suggested the model may not have been relying on the ligand to make that prediction. For the "Add" test, I identified interacting atoms in the same way but bonded a methyl group to the atom instead of changing its atom type. Any created ligands that broke valency rules or were invalid chemically, and so could not be parsed by RDKit, after modification, were excluded from the analysis.

4.3.4 Metrics

Plausibility Metrics

The PoseBusters software was used to check for the plausibility of the ligand-pocket complex (Buttenschoen et al. 2024). The ligand plausibility checks were separated from the ligand-protein checks (i.e. Inter Protein-Ligand Clash and Volume Overlap) to distinguish the causes of physical implausibility. I extended the checks to include the physical plausibility of the side chains individually and to take into account their clashes. To check the individual side chain plausibility, the side chain of each amino acid was parsed through PoseBusters alone to ensure its structure was physically valid. Side chain clashes were considered to occur when two atoms of side chains with VdW radii R_1 and R_2 have a distance (d) that fulfils the inequality used in (Abanades et al. 2023; Miao et al. 2011; Nagata et al. 2012):

$$d < 0.63 \times (R_1 + R_2) \tag{4.3.4.1}$$

Vina Score

The Vina Score was calculated using Smina (Koes et al. 2013), a fork of AutoDock Vina, which has built-in protonation and preparation using OpenBabel (O'Boyle et al. 2011).

4.4 Results

4.4.1 Physical plausibility of ligand pocket generation outputs

One major limitation of the current evaluation of ligand pocket generation methods is that the physical plausibility of the predictions has not been explored. Previous studies (Buttenschoen et al. 2024; Harris et al. 2023) have demonstrated that machine learning methods for related tasks can fall short in predicting physically plausible structures. To address this gap, I curated a set of physical validity tests. Each method generated a new pocket structure with a novel pocket sequence for each protein-ligand complex present in the Astex Diverse set (Astex) and Runs N'

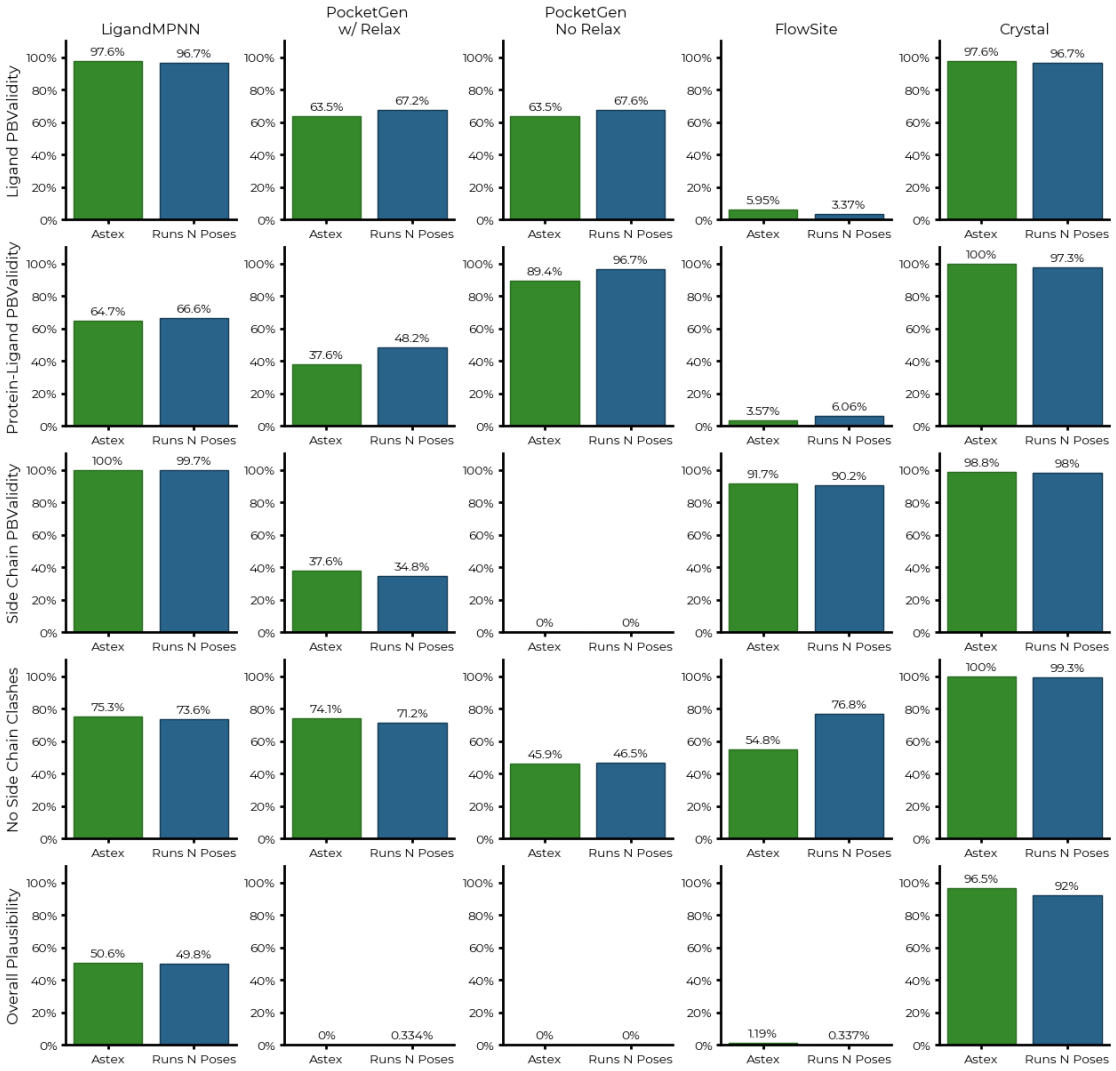
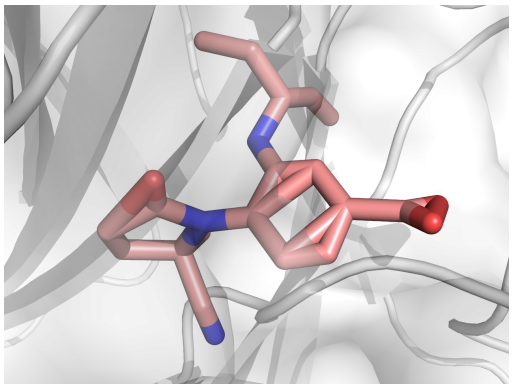
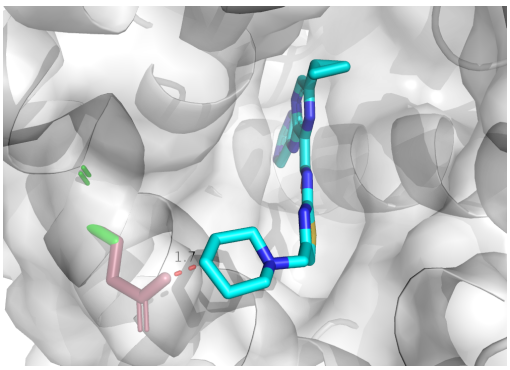


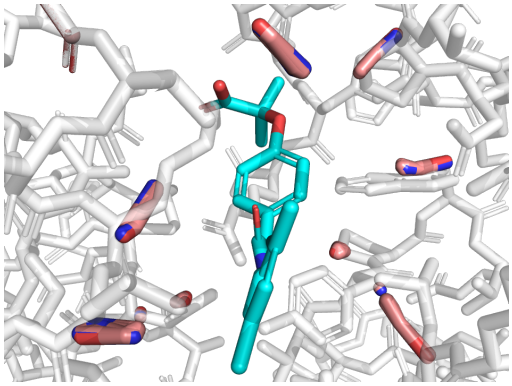
Figure 4.3: Comparison of plausibility metrics across pocket generation methods (LigandMPNN, PocketGen with and without relaxation, Flowsite) and the original crystal structures for the Astex and Runs N' Poses datasets. Each bar plot represents success rates across various metrics, including Ligand PBValidity, Protein-Ligand PBValidity, Side Chain PBValidity, No Side Chain Clashes, and Overall Plausibility. The blue and green bars correspond to the results for the Astex and Runs N' Poses datasets, respectively, with the percentage values above each bar.



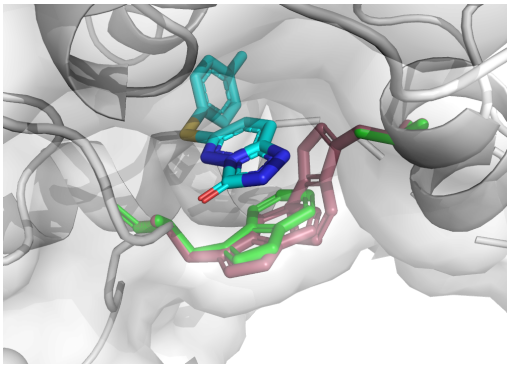
(a) FlowSite ligand prediction for PDB:1VCJ and CCD:IBA with ligand implausibility due to bond lengths and angles. Note PyMol predicts bonds based on distances, hence incorrect bonds between atoms that are too close to each other (red) (Schrödinger, LLC 2015)



(b) LigandMPNN prediction for PDB:5SJU and CCD:K4U with protein-ligand implausibility due to clash. The prediction of a threonine (red) instead of an alanine (green) causes a clash with the ligand (blue).



(c) PocketGen (No Relax) prediction for PDB:1G9V and CCD:RQ3 of side chain (red) and ligand (blue) positions. Unchanged side-chains and backbones are white. Side-chain atoms are clustered in the same regions, leading to severe side-chain plausibility.



(d) LigandMPNN prediction for PDB:7B4X and CCD:SWT of side chains (red) compared to ground truth side-chains (green). Prediction of two tryptophans instead of one tryptophan and alanine results in severe side-chain clashes due to a lack of freely available space to pack the bulky side-chains into.

Figure 4.4: Examples of physical implausibility in pocket design models. (a) FlowSite ligand implausibility, (b) LigandMPNN protein-ligand clash, (c) PocketGen side-chain packing implausibility, (d) LigandMPNN side-chain clashes. 2D structures of the molecules are found in the Appendix (C.2).

Poses (Runs N Poses) (see 4.3.1 in Data and Methods). To measure the physical plausibility of each generated ligand-protein complex, the protein-ligand plausibility tests from the PoseBusters suite were utilised. The ligand-only tests were separated from those involving protein-ligand plausibility (Volume Overlap and Inter-Protein-Ligand Clash). The ligand plausibility test (Figure 4.3 Ligand PBValidity) checks if the bond lengths and bond angles are within thresholds determined to be sensible for a conformer. LigandMPNN performs identically to the crystal structures on this test as it does not modify the input crystal pose. PocketGen and FlowSite both have a lower Ligand PBValidity as both update the ligand coordinates in Cartesian space during the design of the protein pocket. However, PocketGen uses the initial physically plausible crystal pose as a starting point, unlike FlowSite, potentially explaining its higher relative Ligand PBValidity.

The protein-ligand physical plausibility checks (Figure 4.3 Protein-Ligand PBValidity) tests capture whether the ligand and protein are clashing or overlapping in their volumes (see Figure 4.4b). Only PocketGen (No Relax) has close to crystal physical plausibility in both the Astex and Runs N Poses datasets. PocketGen (No Relax) creates side-chain conformations with atoms placed close to the Cβ atoms so they are unlikely to clash with the ligand (see Figure 4.4c). If the relaxation step is run on PocketGen, most protein pocket-ligand complexes fail these tests, as the relaxation is performed without regard for the ligand. The poor pass rate of all methods demonstrates that increased Protein-Ligand validity requires more explicit training or constraining of the outputs of the models.

The inability of PocketGen, without the relaxation step, to generate plausible side chain atomic structures is demonstrated by our side chain validity test, Side Chain PBValidity (Figure 4.3 Side Chain PBValidity). It is also shown in Figure 4.4c. For this test, I passed each side chain, ignoring the backbone coordinates, within the pocket individually through PoseBusters. I ignored amino acids that had no side chain (glycine) or had only one side chain atom (alanine). If any side chain was invalid, the whole protein-ligand complex failed these tests. LigandMPNN, which uses LigandMPNNSCPacker to predict side-chain orientation after sequence

prediction, and FlowSite almost always pass this test as they predict only the chi angles of the side chains. These side chain atomic structures are then generated using the OpenFold package (Ahdritz et al. 2024), guaranteeing physical plausibility. PocketGen, even after relaxation, does not always generate plausible side chain conformations, demonstrating the difficulties of producing physical plausibility for molecular generation in Cartesian space.

The final test is whether there are any side chain clashes within the protein pocket (Figure 4.3 No Side Chain Clashes). The chosen VdW clash threshold for two atoms is commonly used for assessing side chain packing methods (see Data and Methods (4.3.4)) (Abanades et al. 2023; Miao et al. 2011; Nagata et al. 2012). Like in the above Side Chain PBValidity test, if any clash between any side chain exists, the entire protein-ligand complex is considered to fail the test. An example of these clashes for LigandMPNN is depicted in Figure 4.4d. PocketGen demonstrates a similar trend, with the relaxation step improving the pass rate of this test, as relaxation should adjust its orientations to reduce these unenergetically favourable clashes. Interestingly, there is a large difference between the performance of FlowSite on Astex (54.8%) and Runs N Poses (76.8%). This could be due to FlowSite being less able to dock accurately out of distribution, and so not packing the side chains close to the ligand for the Runs N Poses protein pockets. My results show that these methods still struggle to create pockets with good side-chain packing that avoids clashes.

Finally, the tests were combined into a single one (Figure 4.3 Overall Plausibility) with a protein pocket-ligand complex only passing the test if all four previous tests are passed (Ligand PBValidity, Protein-Ligand PBValidity, Side Chain PBValidity and No Side Chain Clashes). PocketGen (No Relax and w/ Relax) and Flowsite are almost never able to produce any plausible protein pocket-ligand complexes for either the Astex or Runs N Poses sets. LigandMPNN produces the most plausible structures, but with an almost 50% success rate for both datasets. However, of all the methods, its co-generation structural prediction capabilities (see Table 4.1) are the most limited, showing that success is most likely arising by not altering much

of the input crystal structure, such as ligand position.

Although these tests might be considered stringent, the native crystal structures almost always pass all of them. Any failures might be due to slight modelling errors when fitting to the electron density. I believe that these tests are useful to assess the plausibility of generated protein pocket-ligand complexes. These tests are not exhaustive, as I have not considered the validity of the backbone, which PocketGen remodels, nor whether the side chain chi angles are within ranges found in crystal structures for a given amino acid.

4.4.2 The confounding effect of physical plausibility on analysis of ligand pocket generation

Physical plausibility is crucial, as generations that violate known experimental rules, such as exhibiting steric clashes, fail to reflect real-world behaviour and are often considered unreliable. However, I demonstrate here that this failure is not only undesirable in itself, but also confounds downstream analyses, skewing prioritisation of structural hypotheses to verify experimentally. To demonstrate this, I considered the Vina scoring function that has frequently been used to score the quality of generations for methods, both for pocket design (Zhang et al. 2024b; Noske et al. 2023) and for other methods (Schneuing et al. 2024; Ziv et al. 2025). While Vina's predicted binding energies are not quantitatively accurate, they serve as a useful and interpretable proxy for assessing the energetics of protein-ligand interactions. However, because Vina scores are approximately a linear function of per-atom energy contributions, they effectively reward larger numbers of interacting atoms. To exploit this, I developed a baseline model (VolumeBaseline), described in 4.3.2, that improves the Vina score by simply increasing the number of protein atoms, thus generating more favourable protein-ligand interactions. As the Vina scoring function does not penalise the increasing number of protein-protein clashes, this strategy can artificially improve scores, as shown in Table 4.3.

VolumeBaseline has the highest Vina scores, exceeding the energy of the original crystal structures. PocketGen (w/ Relax) and FlowSite produce high positive, and

Ligand Pocket Generation Method	**Astex (kcal/mol)**	**Runs N Poses (kcal/mol)**
LigandMPNN	-6.01 ± 0.71	-5.71 ± 0.58
PocketGen (w/ Relax)	19.26 ± 8.97	7.76 ± 2.64
PocketGen (No Relax)	-4.55 ± 4.50	-8.92 ± 0.60
FlowSite	173.93 ± 21.03	173.16 ± 13.66
VolumeBaseline	$\underline{\mathbf{-9.61 \pm 0.55}}$	$\underline{\mathbf{-9.84 \pm 0.43}}$
Crystal	-8.18 ± 0.43	-8.23 ± 0.28

Table 4.3: Comparison of mean Vina Scores across different ligand pocket generation methods for the Astex and Runs N Poses benchmark sets. Significantly lowest (and therefore the better) energy values per dataset are bold and underlined. Error ranges denote the 95% confidence intervals.

so unfavourable, scores as they generated side-chains that often severely clash with the ligand (see Figure 4.3 Protein-Ligand PBValidity). PocketGen without the MD minimisation step (PocketGen (No Relax)) has more negative scores on average, showing that its poor side-chain structural plausibility (see Figure 4.4c) also appears to artificially boost the number of protein-ligand interactions.

Therefore, to reliably use the Vina score to score generated structures requires the exclusion of physically invalid complexes. In doing so, I show that almost all methods, except LigandMPNN (Astex = 9.4% and Runs N Poses = 9.7%), are unable to generate complexes that exceed the Vina score of the crystal structure whilst still being valid, using the tests I have curated (see Table 4.4).

Overall, this demonstrates that if Vina is going to be used to validate ligand pocket generation methods or to prioritise experimental validation, scores must only be considered after removal of physically implausible poses. Furthermore, existing methods cannot reliably co-generate a pocket sequence and structure that exceeds the existing crystal structure's Vina score, showing the need for improvement in the field.

4.4.3 Beyond amino acid recovery: benchmarking amino acid predictions using deep mutational scanning data

So far, I have demonstrated that the structures generated by ligand pocket generation methods are limited in their physical plausibility, which confounds downstream analysis. However, the predicted sequence for the pocket can be analysed alone.

Method	All	Valid
Astex		
LigandMPNN	12.9%	**9.4%**
PocketGen (Relax)	27.4%	0.0%
PocketGen (No Relax)	69.4%	0.0%
FlowSite	0.0%	0.0%
Volume Baseline	**92.9%**	0.0%
Runs N Poses		
LigandMPNN	15.1%	**9.7%**
PocketGen (Relax)	34.4%	0.0%
PocketGen (No Relax)	72.6%	0.0%
FlowSite	0.0%	0.0%
Volume Baseline	**92.3%**	1.0%

Table 4.4: Percentage of poses with Vina scores (kcal/mol) more negative than crystal structures. Results are shown separately for Astex and Runs N Poses benchmarks. Success is distinguished by whether just being more negative than crystal structures (All) and being more negative and physically plausible according to tests in Figure 4.3 (Valid). Highest percentages are shown in bold.

Typically, in the development of these methods, the recovery of the original ground truth sequence, or simply the amino acid recovery, is used to evaluate model performance.

I first examined the mean amino acid recoveries for the Astex and Runs N Poses datasets, displayed in Figure 4.5. The results show that FlowSite performs the worst with very low amino acid recoveries and an 11.6% difference between the datasets (Astex represents an in-distribution dataset and Runs N Poses represents an out-of-distribution dataset), demonstrating poor generalisation. There is no significant difference between LigandMPNN (66.2±2.7%) and PocketGen (63.3±6.2%) for the Astex set. PocketGen does not generalise in its accuracy for Runs N poses with a difference of 18.9%, indicating it has potentially overfit to its training data. LigandMPNN has the smallest difference of 7.2% between the two datasets. Based on these results, LigandMPNN appears to be the most effective methodology. However, amino acid recovery is a limited metric for performance as it penalises

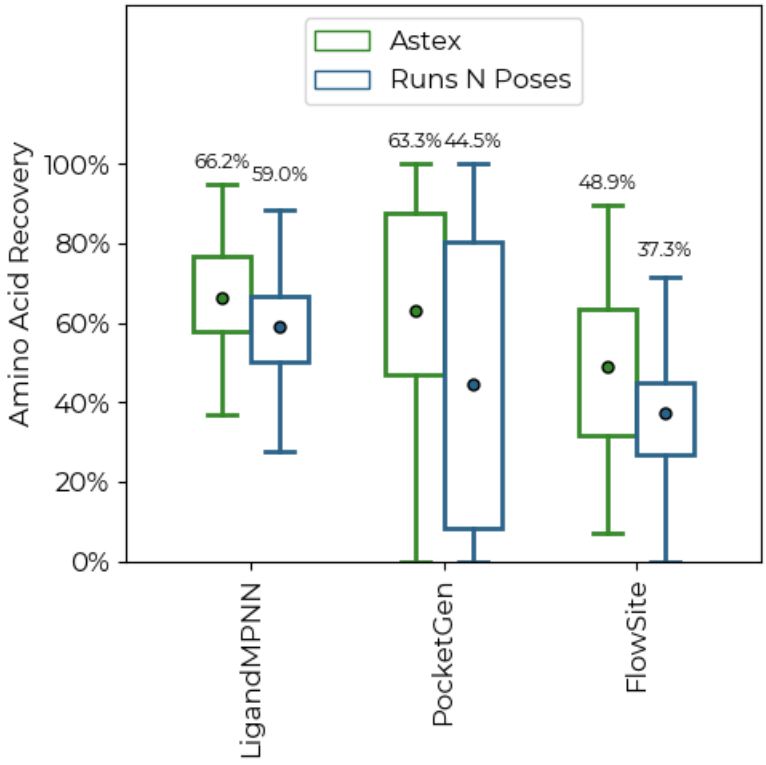


Figure 4.5: Boxplots of the distribution of amino acid recoveries for generated ligand-pockets for Astex and Run N Poses datasets. The mean amino acid recoveries are depicted as dots, with values displayed above the tails. The median values are not represented.

novel and useful predictions that do not match the original sequence. A generated pocket sequence with 95% similarity to the original sequence could be unable to bind the specified ligand, whilst a completely novel sequence could.

To address this limitation, I propose a new benchmark that measures performance based on the deep mutational scanning data for the MET kinase for six different inhibitors (Estevam et al. 2024). The MET Kinase is a tyrosine kinase receptor that is a key drug target for cancers due to its activation contributing to tumour growth and spread. This analysis is limited to this singular data source, as I could not find any other datasets in the literature that performed deep mutational scans for a protein against multiple ligands. Deep mutational scanning experiments systematically measure the impact of every or nearly every possible mutation of a protein on its function, binding to another molecule, or another property, using a high-throughput assay. In this work, the authors measured the binding of the

inhibitors by quantifying cell survival after inhibition and compared it to a DMSO control. I processed these raw sequence reads to classify each mutation's effect individually (described in 4.3.1 and visualised in Figure 4.2) as either Neutral, Inhibitor Enhancing, Protein Destabilising, Inhibitor-induced Protein stabilising, Unknown (if not assayed) and WT (the original amino acid). Note this benchmark assumes that the impact of each mutation was independent of any other, a flawed but necessary assumption. I measured the relative enrichment of that class of prediction's mutation in a sequence compared to randomly sampling that type of mutation. Each method predicted ten sequences for each PDB corresponding to the inhibitor-MET kinase complex; their pocket sequences were separated into their individual pocket residues and mapped to the mutation needed to produce this pocket residue. The mutation's classification was therefore given to the predicted pocket residue. To standardise the probability of each mutation class, I divided each by the probability of randomly picking that mutation class in a sequence. Figure 4.6 shows that all three of the methods do not enrich for any mutation type except for WT (the original amino acid). This result indicates that although the methods can recapitulate the original sequence, they cannot successfully explore any other class of mutation better than random. Therefore, it is less likely that the models are generating sequences highly similar to the original sequences because they create a strong binding pocket for a ligand with appropriate mutation classes. Instead, the models might have learnt simply to repredict the original sequence only. This trend would not be apparent using analyses of these methods that relied on only the amino acid recovery metric.

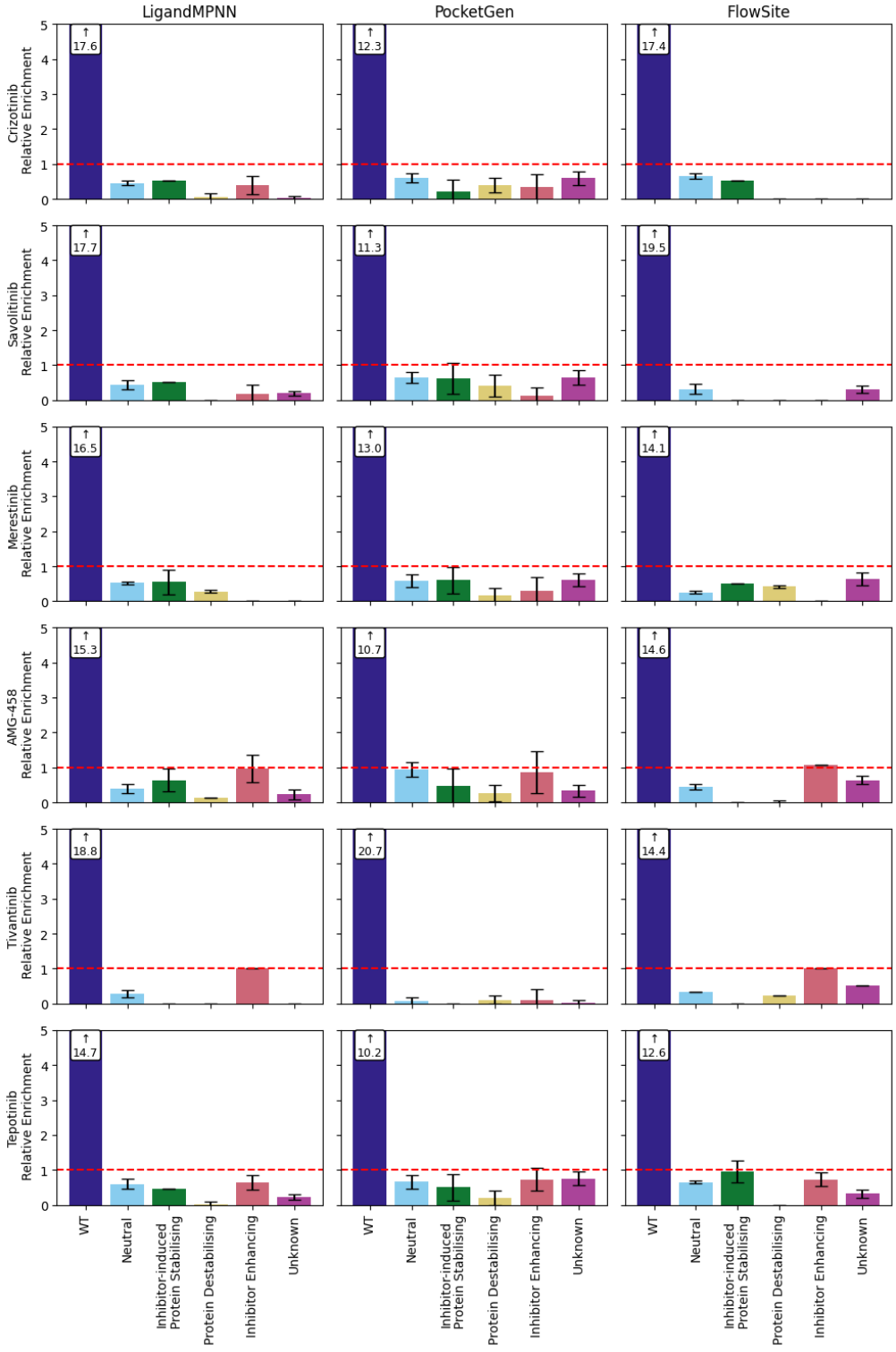


Figure 4.6: Relative enrichments of mutation classes, classified based on deep mutational scanning data from (Estevam et al. 2024) for the MET Kinase for different ligand-pocket generation methods (LigandMPNN, PocketGen and FlowSite). An enrichment of 1, which is the probability of a class of mutation being picked relative to the probability of randomly picking that mutation, is indicated as a red dashed line.

4.4.4 Adversarial ligand change tests

Given that the evaluated models fail to reliably propose meaningful sequences that explore anything more than sequences close to the original sequence, I sought to test whether their predictions are sensitive to the ligand's local environment at all. This sensitivity is an important property of a model as it demonstrates that it has learnt the causal relationship between residue prediction and the local ligand environment. The methodology for these tests consisted of taking each protein-ligand complex from the Astex and Runs N Poses benchmark sets and predicting a novel sequence. These generated protein pockets had their interactions calculated with the ligand using the interaction profiler PLIP (Salentin et al. 2015). For each interaction that was either a hydrogen bond or hydrophobic, the ligand atom type for the interacting atom was changed to one that would be unable to fulfil that interaction. For example, an oxygen that forms a hydrogen bond with a residue would be changed to a carbon, and so no longer able to form that hydrogen bond. The rate of change of that residue after ligand alteration ("Mutation Rate") was calculated as a measure of sensitivity. The methodology is further detailed in 4.3.3. As FlowSite often fails to dock the ligand into the pocket it is redesigning, I excluded it from this analysis. For the "Swap" test (Figure 4.7), LigandMPNN displayed some sensitivity to local changes in the ligand with its residue predictions. The background mutation rate, shown in a faded colour in Figure 4.7, represented the residue change when another non-interacting ligand changed. LigandMPNN displays local sensitivity as the mutation rate for both datasets is higher than the background mutation rate for both hydrogen bond and hydrophobic interactions. It is more sensitive to ligand changes for hydrogen bonds than for hydrophobic interactions. This reflects the promiscuous nature of hydrophobic interactions and so the reduced likelihood that a residue prediction is important for an interaction (Errington et al. 2025). PocketGen, however, demonstrates almost no sensitivity to local ligand change that prevents interactions or any ligand changes at all.

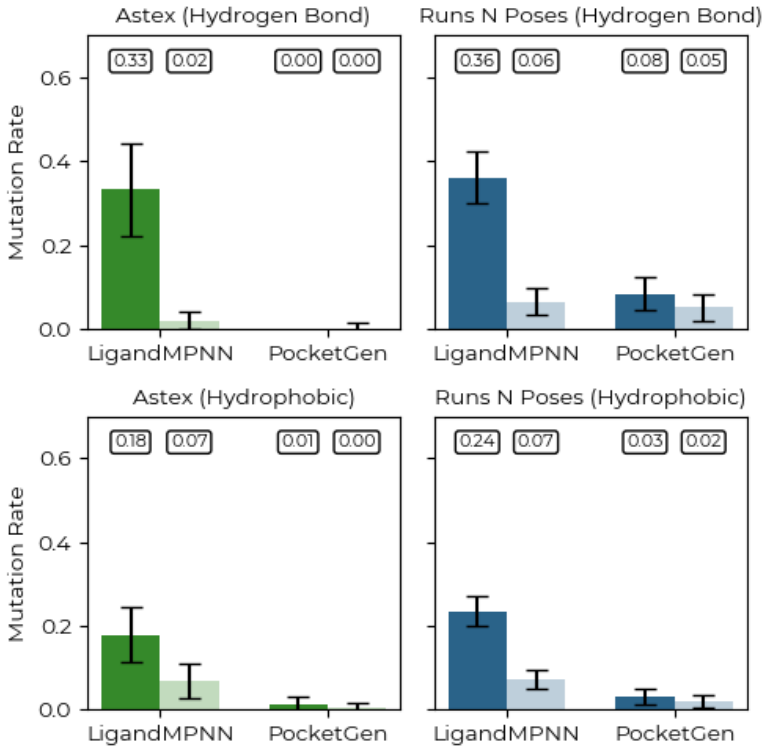


Figure 4.7: Bar plots for the mutation rate for LigandMPNN and PocketGen for hydrogen bond and hydrophobic-forming amino acid predictions after adversarial ligand atom type swap. Changes are shown in green and blue for Astex and Runs N Poses datasets, respectively, with faded colours representing background mutation rate after non-local ligand atom swap. Error bars represent the 95% confidence intervals on the mean mutation rate. The mean mutation rate is displayed in boxes above each bar.

As a further test, instead of replacing the atom type, I added a methyl group ("Add" test) to the interacting ligand atom. This is a more significant ligand change as the number of atoms will change, and the methyl group is likely to be sterically clashing with the interacting residue or nearby residues. The results, shown in Figure 4.8, reinforce that PocketGen is not sensitive to local ligand changes and so is unlikely to be making residue predictions based on forming favourable protein-ligand interactions. LigandMPNN showed improved sensitivity compared to the "Swap" test for both interaction types and datasets. The background mutation rate also increased, reflecting that the methylation had a less local impact on amino acid prediction for the pocket. These results expose a stark difference in method performance, with PocketGen unresponsive to ligand changes. The reason behind this could be either PocketGen's use of ESM2 embeddings in its structural encode

module to improve amino acid recovery, causing the model to become reliant on that information for prediction instead of the ligand itself. Another possibility is that LigandMPNN only considers each amino acid prediction in its local environment, which encourages the model to use the ligand information as part of its prediction.

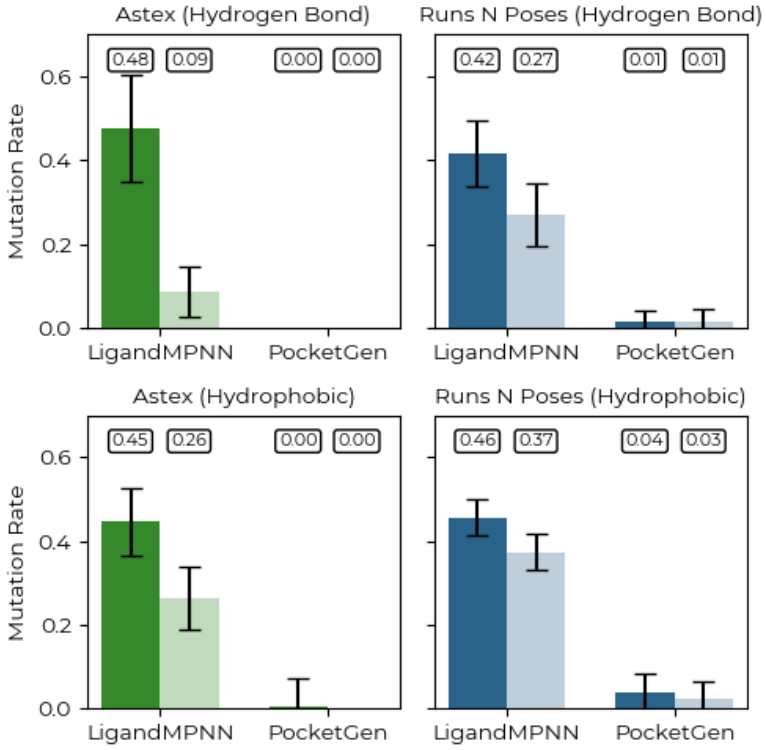


Figure 4.8: Bar plots for the mutation rate for LigandMPNN and PocketGen for hydrogen bond and hydrophobic-forming amino acid predictions after adversarial ligand atom type methylation. Changes are shown in green and blue for Astex and Runs N Poses datasets, respectively, with faded colours representing background mutation rate after non-local ligand methylation. Error bars represent the 95% confidence intervals on the mean mutation rate. The mean mutation rate is displayed in boxes above each bar.

4.5 Discussion

Overall, I have set out in this chapter to establish the feasibility of using ligand pocket generation methods to expand and balance the existing structural data used to train structure-based machine learning models. In doing so, I have established tests that have probed their performance and identified the pitfalls of the methods. These results should help guide the development of future ligand pocket generation

methods.

First, current methods that co-generate a sequence for a pocket and a structure are not able to generate sufficiently high-quality structures for downstream analysis. The methods examined, LigandMPNN, PocketGen and FlowSite, have varying structure generative capabilities (see Table 4.1), but none can consistently generate physically plausible structures, measured by my curated physical plausibility tests. Furthermore, I have demonstrated that this physical plausibility confounds the downstream analysis of these generated pockets. By developing a simple baseline method that predicts the largest possible amino acids by volume that do not clash with the ligand, I showed that achieving higher Vina scores than any method is possible. In doing so, the side-chains of the predicted amino acids often clash with each other and so are highly physically implausible. In this work, we considered the Vina scoring function alone; however, physical plausibility would likely affect any other tool's analysis of generated pockets. Most tools and metrics, such as pocket pharmacophoric properties and energy calculations, are developed for physically plausible structures. Therefore, these methods' co-generated structures should not be analysed themselves; instead, they should be re-predicted using fast and accurate co-folding methods that can predict the protein and ligand with high physical plausibility. I explore this idea in the next chapter.

Furthermore, this work explored the ability of these methods to predict binding sequences to a ligand. By using a DMS dataset (Estevam et al. 2024) for the MET Kinase binding to six different inhibitors, I show that methods can recapitulate the original sequence of a given pocket and ligand, but they cannot accurately or usefully explore other sequences that could enable binding. It is not surprising that these methods are unable to recapitulate the ligand-sequence relationships to succeed at the DMS benchmark, as they have not been trained on data of this type. Instead, they have been trained on the very data I aim to expand. Furthermore, by adversarially testing the local responsivity of predictions to ligand changes, I was able to identify that PocketGen appears to make predictions independently of the ligand. Despite LigandMPNN performing similarly to PocketGen on the

deep mutational scanning benchmark, it is more responsive to ligand changes and so appears to make predictions based on the input ligand. Therefore, currently, these methods are unlikely to be helpful in generating synthetic structural data for training other ML methods. Therefore, currently, these methods are unlikely to be helpful in generating synthetic structural data for training other ML methods. For methods to improve and learn more than the recapitulation of the original sequence, a different paradigm for developing methods is needed that does not rely on amino acid recovery as a loss function. I explore the possibility of using structure prediction hallucination instead to design sequences in the next chapter. These benchmarks and the pitfalls they have identified could drive progress in the field. The software for this analysis is available at https://github.com/guydurant/pauge and implemented as a command-line tool, Pauge, to gauge the quality of generated pockets.

5

Do co-folders dream of synthetic protein-ligand complexes?

Contents

5.1 Preface

In the previous chapter, I found that ligand pocket generation machine learning methods struggle to generate physically plausible structures as part of their prediction, limiting their applicability for downstream analysis. Further, they do not explore ligand pocket sequences successfully and appear to have overfit to mostly recapitulating the original sequence of a pocket.

In this chapter, I begin to address both problems by developing a methodology to improve the speed of co-folding methods. Current co-folding methods are generative diffusion models able to co-predict the protein structure and the ligand conformer simultaneously by denoising atomic coordinates. However, their inference speeds scale quadratically with the number of atoms or tokens in the protein-ligand complex, reducing their throughput. To overcome this, I developed a diffusion guidance method, called ScrewzFix, that predicts the protein pocket alone bound to the ligand and guides the protein residues of the pocket to the positions in a provided complete structure, using the Boltz-1x model (Wohlwend et al. 2025). This reduces the inference times required to predict how the protein and ligand interact simply by reducing the size of the system that the co-folder needs to predict. Further, I examined the ability of using this method for ligand-conditioned side chain packing for the accurate prediction of pocket conformations with high physical plausibility for a given ligand position, backbone position, and pocket sequence.

This inference speed-up enabled the development of a hallucination-based ligand pocket sequence generation method, called Sparkz, which optimises pocket sequences based on the confidence metrics of Boltz-1. Hallucination methods, such as BindCraft (Pacesa et al. 2024), have shown success in creating protein binders to specified epitopes, but have not been successfully translated to the design of proteins to bind other modalities, such as small molecules. Sparkz successfully optimises for a simple confidence score to hallucinate pocket sequences, but it is not clear if these sequences would bind the molecules in vivo. However, this work is preliminary and would benefit from further development and benchmarking. This incomplete

work acts as a small step towards developing the pipeline for generating synthetic structural protein-ligand training data described in the previous chapter.

5.2 Introduction

The accurate prediction of biological and chemical structure from sequence is an exciting advancement in the field of computational biology. The first breakthrough, AlphaFold 2 (Jumper et al. 2021), was revolutionary in combining key advancements such as learning structural priors from MSAs and using IPA to predict protein and protein complex structures. However, the need for prediction of other modalities beyond just protein, such as chemical ligands, DNA and carbohydrate groups, has encouraged the development of "co-folding" methods that can predict these biological complexes with high accuracy (described in detail in Section 1.5.7). To enable the prediction of these different classes of molecules, diffusion models have been adopted that denoise the complexes' atomic positions from Gaussian distributions to generate the structure. AlphaFold 3 (Abramson et al. 2024) was the first method to employ this approach, with subsequent open-source methods developed that aimed to match its accuracy (ByteDance et al. 2025; Wohlwend et al. 2025; Passaro et al. 2025; Chai-Discovery-Team et al. 2024). These methods utilise the MSA of the protein sequence, like AlphaFold 2, but also attend to additional tokens of other modalities to build up rich representations for each atom or token of the biological complex. An atomistic diffusion model then iteratively updates each token's or atom's coordinates from random positions so that they are denoised to the correct positions. In this work, I utilised the open-source Boltz models, specifically the Boltz-1x (Wohlwend et al. 2025). Boltz-1 employed a similar architecture to AlphaFold 3, detailed in Figure 1.16, but differs in that it passes all the intermediate denoised structures into the confidence head instead of just the final. By using steering and guidance through the denoising process, Boltz-1x, its successor, was able to fix the physical validity problems of the original model. Boltz-1 can accurately predict the backbone, side-chain orientation, and ligand positions of protein-ligand complexes, making it a valuable method for analysing the structures of generated ligand pockets, such

as those generated in the previous chapter. During the development of this work, Boltz-2 was released (Passaro et al. 2025), which also predicts binding affinities for small molecule-protein complexes with faster inference speeds; however, due to its training data's later date cutoff (June 1, 2023), the method could not be directly benchmarked against the benchmarks of the previous chapter.

De novo protein structure generation can be trained from scratch using diffusion and flow matching to design novel backbones and sequences, as discussed in the previous chapter (Watson et al. 2023; Yim et al. 2023a). However, these methods rely on training large models, which are computationally expensive and require extensive filtering using structure prediction methodologies that have been found to predict experimental success (Watson et al. 2023; Bennett et al. 2023). Therefore, researchers have developed an alternative methodology for protein design that directly utilises this predictive accuracy for binder filtering. Referred to as structure-based hallucination, these methods exploit these structure-prediction tools, used for filtering, to iteratively predict structures and optimise the sequence based on metrics or their confidence predictions of the structure (Wicky et al. 2022; Anishchenko et al. 2021; Goverde et al. 2023). Older implementations relied on Monte Carlo search of the protein sequences, whilst keeping the sequence discrete and optimising through discrete jumps or mutations in the sequence. This process was found to be slow and ineffective due to the speed of structure prediction holding back performance. To overcome this, the discrete sequence distribution used for the structure model can be relaxed to be a continuous probability distribution over the possible amino acids (Goverde et al. 2023). Therefore, the predicted confidence in the structure can be backpropagated through the weights of the structure predictor, thereby optimising the sequence representation, enhancing the confidence of the predicted structure, and the sequence probability distribution can be compelled to converge to being discrete during this optimisation. This convergence ensures that the final output sequence distribution is close to a one-hot encoding of a sequence, and so a single sequence. Experimental success has come with methods such as BindCraft (Pacesa et al. 2024), which has been successfully applied to protein

binder design to 12 diverse targets and came in first place in the first iteration of the protein binder competition run by AdaptyvBio (Cotet et al. 2025). Extending this protein binder success to just ligand-binding pockets, using cofolding instead of AlphaFold 2, could enable the design of novel enzymes, biosensors and signal transducers. Furthermore, it could enhance the expansion of structural data used to train other models by redesigning underrepresented protein-ligand complexes in the PDB. However, a limiting factor for the implementation of such a method is the slow inference time of cofolding methods, which scales poorly with the size of the system. Ligand pockets can be part of large proteins or interfaces with multiple chains, requiring predictions of large complexes to generate the full pocket. During the development of this work, hallucination has been applied to Boltz-1 for protein binder design for small molecules and other modalities, BoltzDesign1 (Cho et al. 2025). This method did not optimise the sequence through the structure representation, but instead through the predicted distogram loss and the confidence without gradients flowing through the diffusion module. This approach bypasses the need for structure prediction, enabling tractable hallucination of sequences. However, it can only design completely novel proteins with no restriction on binding location and does not utilise structure-based losses, which could help generate binding pocket sequences.

To enable faster predictions of generated ligand pockets for analysis and hallucination, I developed a simple workaround to speed up co-folding, ScrewzFix. By predicting only the protein residues in proximity to the pocket and guiding their positions to maintain the correct tertiary structure, I was able to predict the protein accurately, despite these unphysiological "trimmed" proteins not existing in the training data nor in nature. I explored the impact of accuracy and strategies to enable accurate ligand docking and ligand-conditioned side chain packing. Finally, by enabling faster inference, I implemented Sparkz, a hallucination method for designing ligand pockets using Boltz-1x. The implementation of these methods is available at the following link: https://github.com/guydurant/sparkz.

5.3 Data and Methods

5.3.1 Benchmarks and data

Crystal structures for the Astex Diverse set (Astex) and Runs N' Poses benchmark sets (Runs N Poses) were curated as described in Section 4.3.1. The MET kinase deep mutational scanning dataset was also developed as described in Section 4.3.1, with performance of other ligand-pocket methods taken from Chapter 4.

5.3.2 Metrics

All accuracy and physical validity of protein-ligand complexes were measured using metrics as described in Chapter 4.

5.3.3 Protein complex "trimming"

To reduce the number of tokens needed to predict a protein-ligand complex, only residues within 8Å of a specified ligand input were retained. If the picked residues were consecutive in sequence, I grouped them into a new chain for inference. To reduce the total number of chains, and so the number of MSAs needed to be calculated using the MMSeqs2 server (Steinegger et al. 2017), I included any residue within 10 residues on either side of each new chain's original sequence. This expansion step resulted in the trimmed chains being combined into fewer, larger chains. Non-protein atoms that were not the ligand of interest were included if within 8Å of the ligand, except if part of DNA or RNA. Figure 5.1 illustrates the trimming of the protein complex to reduce the number of tokens predicted.

5.3.4 Boltz-1x and atomic guidance

As detailed in Section 1.3.7, the forward diffusion process can be described as a Markov Chain that is gradually perturbing data, in this case, atomic coordinates:

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}\Big(\mathbf{x}_t; \sqrt{1-\beta_t}\,\mathbf{x}_{t-1},\, \beta_t \mathbf{I}\Big), \quad t = 1, \ldots, T, \tag{5.3.4.1}$$

where $\mathbf{x}_0 \in \mathbb{R}^{3N}$ represents the atomic coordinates of a biological complex of N atoms, drawn from an empirical data distribution $p_{\text{data}}(\mathbf{x})$. $\{\beta_t\}_{t=1}^{T}$ is a predefined

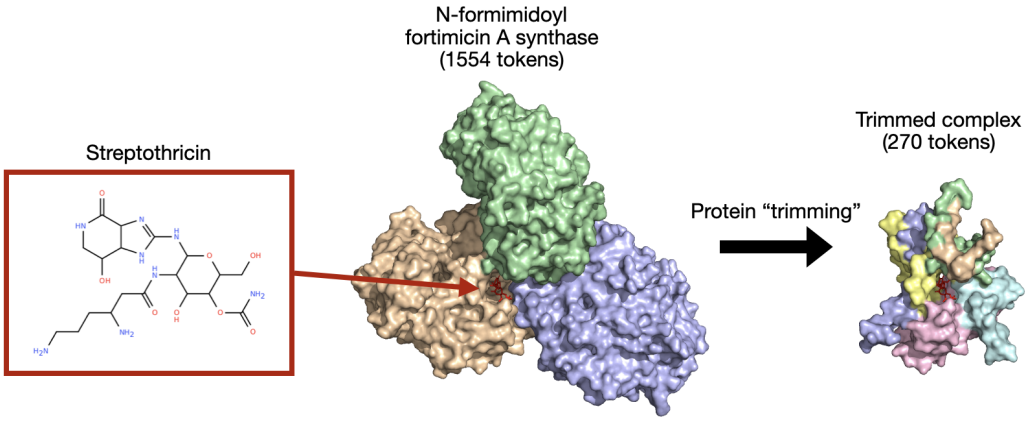


Figure 5.1: Example of trimming a protein-ligand complex, specifically N-formimodoyl fortimicin A synthase (PDB:7XXP) bound to streptothricin. This binding occurs at the interface of three large protein chains (coloured green, yellow and purple). By trimming the residues using the above method, the trimmed complex, consisting of six smaller chains (coloured purple, pink, blue, green, brown and yellow), is ∼5x smaller in token count.

variance schedule. After sufficiently many steps, the marginal distribution converges to the standard normal distribution, i.e. $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

Boltz-1x, parameterised by θ, is trained to reverse this noising to generate an accurate prediction of a biological complex's atomic coordinates. At timestep t, the model receives a noisy atomic configuration $\mathbf{x}_t$ and predicts an estimate of the noise added in the hypothetical forward step $\hat{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t, t)$. Denoising proceeds by updating $\mathbf{x}_{t-1}$ by removing this noise, but adding small additional noise, $\mathbf{z}$, to introduce stochasticity in the sampling process.

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{1-\beta_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sigma_t}\,\hat{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t, t)\right) + \sqrt{\tilde{\beta}_t}\,\mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{5.3.4.2}$$

with $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, $\sigma_t = \sqrt{1-\bar{\alpha}_t}$, and $\tilde{\beta}_t = \beta_t \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}$.

The authors of Boltz-1x employed two complementary mechanisms to bias generation to improve physical plausibility: (i) *guidance*, where potentials $U(\mathbf{x})$ modify the predicted next step with weighting γ,

$$\mathbf{x}_{t-1}^{\text{guided}} = \boldsymbol{\mu}_\theta(\mathbf{x}_t, t) + \gamma\,\frac{\beta_t}{\sqrt{1-\beta_t}}\,\nabla_{\mathbf{x}_t} U(\mathbf{x}_t) + \sqrt{\tilde{\beta}_t}\,\mathbf{z}, \qquad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{5.3.4.3}$$

with

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{1-\beta_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sigma_t}\,\hat{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t, t)\right) \tag{5.3.4.4}$$

and (ii) *steering*, where proposals are reweighted in a Sequential Monte Carlo framework using Boltzmann factors, $\exp(-\lambda E(\mathbf{x}_t))$, before resampling. Guidance alters the reverse dynamics directly, while steering biases the sample population toward low-energy conformers through importance weighting.

I extend this guidance further by implementing *atomic guidance* to encourage individual atoms to match their noisy ground-truth coordinates:

$$\mathbf{x}_{t-1}^{\text{atomic}} = (1 - w_t)\mathbf{x}_{t-1}^{\text{guided}} + w_t M \mathbf{y}_{t-1} \tag{5.3.4.5}$$

Here $M \in [0, 1]$ is the atom mask, $w_t \in [0, 1]$ is the atomic guidance weight, and $\mathbf{y}_t$ is the target noisy coordinates. The atomic guidance weight decayed smoothly with diffusion progress $p_t \in [0, 1]$ according to

$$w_t = \max\left(0,\, 1 - \left(\tfrac{p_t}{0.95}\right)^2\right), \tag{5.3.4.6}$$

where $p_t = t/T$ denotes the normalized timestep. This schedule starts at $w_0 = 1.0$ and decreases quadratically, reaching zero as $p_t \to 0.95$. I chose this to provide a strong correction when noise in the atomic coordinates is high, but gradually relax it to avoid biasing the final low-noise conformations.

5.3.5 ScrewzFix: faster co-folding through protein trimming and atom guidance

By combining the atomic guidance to specified ground truth and the trimming of the proteins, I implemented three configurations of ScrewzFix, a method for fast prediction of bound ligand pockets with limited backbone flexibility: RigidDock, FlexDock and FlexSCPack. The FlexDock configuration enables Boltz-1 to predict unguided side chain positions within 5Å of the ligand pocket, whilst RigidDock does guide them. The FlexSCPack configuration, for ligand-conditioned side chain packing, requires a ligand position to be defined and guides both the backbone and the ligand positions. Like FlexDock, side-chains within 5Å of the ligand pocket are predicted unguided. The process for FlexDock is depicted in Figure 5.2. The differences in the different configurations are outlined in the table below (Table 5.1).

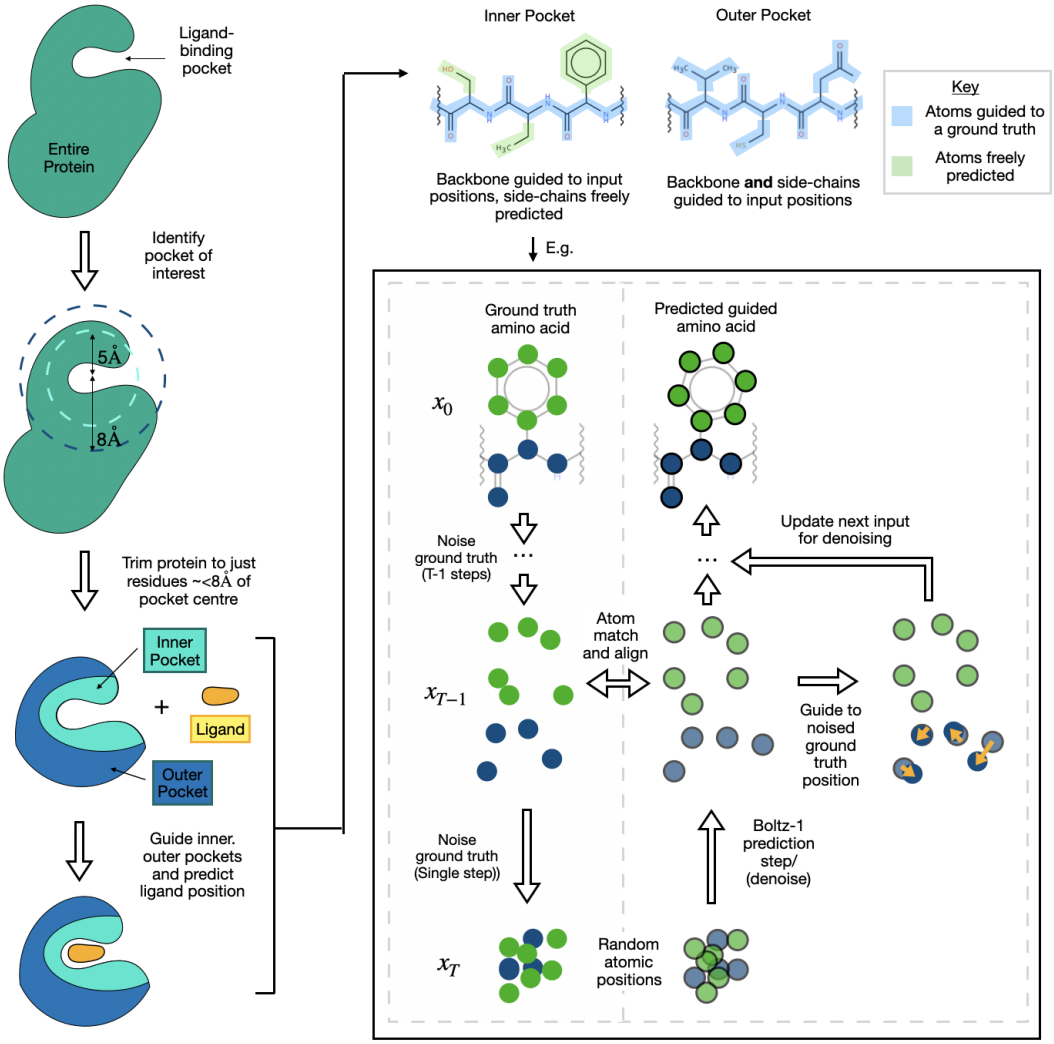


Figure 5.2: Depiction of the process for inference using ScrewzFix (FlexDock). First, the protein is trimmed based on residues within cutoffs of the ligand pocket. The "Inner Pocket" consists of residues within 5Å of the ligand, and the "Outer Pocket" consists of those 5-8Å and within a sequence buffer of these residues. The Inner Pocket has backbone positions guided, but not side-chain positions, for FlexDock, and the Outer Pocket has all atoms guided to ground truth positions. This guidance is depicted, inset, with guided atoms in blue and unguided in green.

In addition to the atomic guidance, I implemented a physical plausibility guidance term to improve the physical plausibility of the side-chains and to avoid clashes, I used the popular threshold for side-chain clashes (Abanades et al. 2023; Miao et al. 2011; Nagata et al. 2012) where a clash occurs when two atoms with VdW

Table 5.1: Overview of different ScrewzFix configurations and their guidance components.

Configuration	Backbone guided	Ligand guided	Pocket side chains guided
RigidDock	✓	✗	✓
FlexDock	✓	✗	✗
FlexSCPack	✓	✓	✗

radii R_1 and R_2 have a distance (d) that fulfils this inequality:

$$d < 0.63 \times (R_1 + R_2) \tag{5.3.5.1}$$

This resulted in the following flat-bottom potential

$$V_{\text{scvdw}} = \sum_{i<j} \max\Big(0,\ \alpha\big(R_i + R_j\big) - d_{ij}\Big)^2, \qquad \alpha = 0.63 \tag{5.3.5.2}$$

This potential was applied at every denoising step for guidance, with a weighting of 0.05 and a resampling weight for steering of 0.10, matching the parameters of the potentials found in Boltz-1x. Inference was performed using MSAs generated for each protein chain. I found in testing that inference without utilising the MSAs was inaccurate, as Boltz-1x heavily relies on their inclusion for accuracy.

5.3.6 Sparkz: structure-prediction hallucination with ScrewzFix

To hallucinate a sequence using a structure-prediction method, in this case Boltz-1x, a structure is predicted from a probabilistic sequence representation rather than a conventional one-hot encoding. This relaxation enables continuous optimisation over the sequence probabilities, rather than slower, discrete optimisation. In my implementation, heavily based on the BindCraft method (Pacesa et al. 2024), optimisation is restricted to residues within 5Å of the ligand pocket, while gradients for all other residues are zero-ed, leaving the rest of the sequence unchanged. The randomly initialised sequence is then predicted with the loss of the chosen confidence metric of the structure backpropagated through Boltz-1x's weights back to the sequence distribution of the pocket. Using the Adam optimiser, with a learning rate of 1, and the gradients of the confidence prediction, this distribution is then optimised

and then re-predicted using the Boltz-1x method, to explore sequence space, as done in BindCraft. After the first 25 steps, ScrewzFix, instead of predicting the optimised sequence probability, predicts the softmax of the sequence probability distribution, with temperature (t) increased according to the following relationship below:

$$t(\mathrm{i}) = 0.01 + 0.99\left(1 - \frac{\mathrm{i}+1}{\mathrm{n}}\right)^2$$

This encourages the optimisation to tend towards a one-hot encoded representation, thereby creating a real pocket sequence. This is done for a further 20 steps. For the final five steps, instead, the confidence is still backpropagated through the $softmax$ sequence representation, but ScrewzFix predicts the $argmax$ of the sequence representation and so one-hot encoded sequence. These three stages match those of BindCraft. However, I dropped the final fourth stage of BindCraft, which is sampling of random discrete mutations, to both simplify and speed up Sparkz. The full Sparkz methodology is detailed in Figure 5.3.

5.3.7 Comparison to other models

Boltz-1x

To limit the computational expense of predicting the entire complexes, I only co-folded any protein chain within 10Å of the crystal ligand pose of interest. Furthermore, I included any inorganic or organic cofactor that was within 5Å of any of these protein chains that was not the ligand of interest for docking. MSAs for protein chains were generated using the MMSeqs2 server (Steinegger et al. 2017), as part of the Boltz package. Only one complex per protein-ligand complex was predicted.

LigandMPNN-SCPacker

LigandMPNN-SCPacker utilises the encoder-decoder GNN architecture used for the inverse folding methods ProteinMPNN (Dauparas et al. 2022) and LigandMPNN (Dauparas et al. 2023). However, the amino acid identity is also given as a feature, and the model predicts a mixture of three circular normal distributions that are

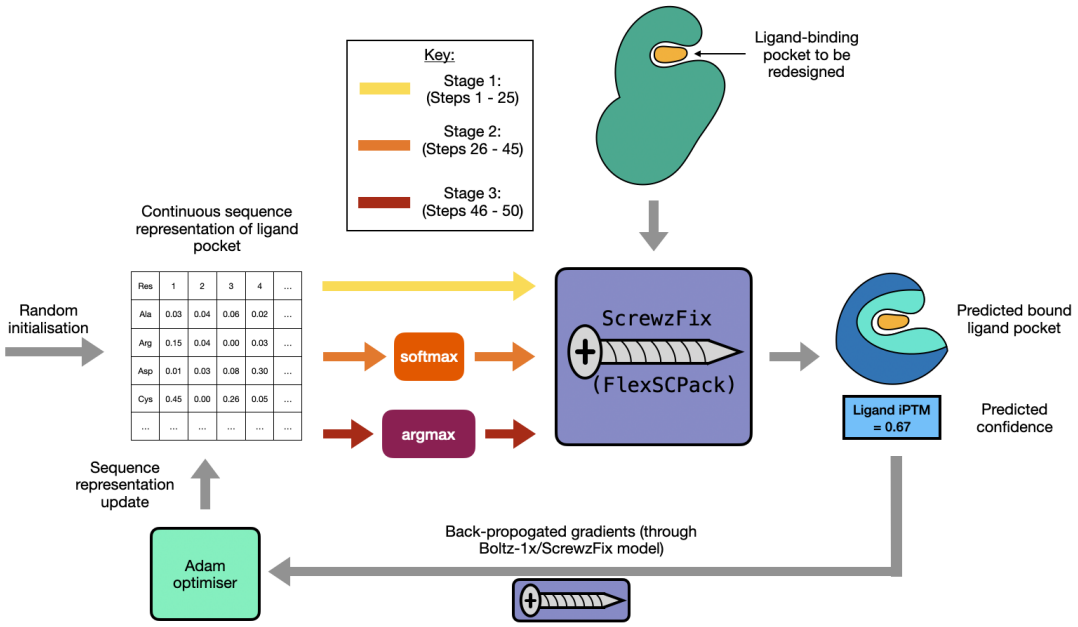


Figure 5.3: Illustration of the hallucination of ligand-pockets using Sparkz. Randomly initialised continuous sequence representations have structures predicted using ScrewzFix (FlexSCPack) with an output confidence. This confidence is backpropagated through the weights of the Boltz-1 model and used to optimise, via the Adam optimiser, the sequence representation. Stages 1-3 have the sequence representation predicted either as it is (1), through applying $softmax$ with increasing temperature (2) and applying $argmax$ (3). The final representation is then used to produce a discrete ligand pocket prediction.

decoded to chi angles from 1 to 4, depending on the number of chi angles the side chain consists of. The atom coordinates that result from each chi angle and the value itself are included in the prediction for subsequent decoding. Inference was run using the provided code (https://github.com/dauparas/LigandMPNN), with default parameters and only one structure generated per ligand pocket.

5.4 Results

5.4.1 Impact of guidance and protein "trimming" on speed and ligand docking accuracy

Co-folding promises high accuracy in predicting protein-ligand complexes by accounting for the conformational changes of both ligands and proteins upon binding. However, its speed of prediction is a limiting factor in its application to throughput

analysis of protein-ligand interactions, such as for off-target prediction, virtual screening and structure prediction hallucination for protein design.

One cause for this slow inference speed is the quadratic scaling of prediction according to the number of tokens, amino acids, or atoms of non-protein molecules that need to be predicted, as depicted in Figure 5.4. This scaling is due to the PairFormer component of the model, which attends to all tokens at once. Here, I

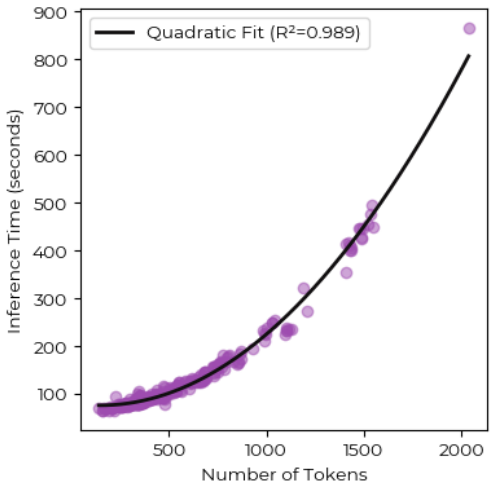
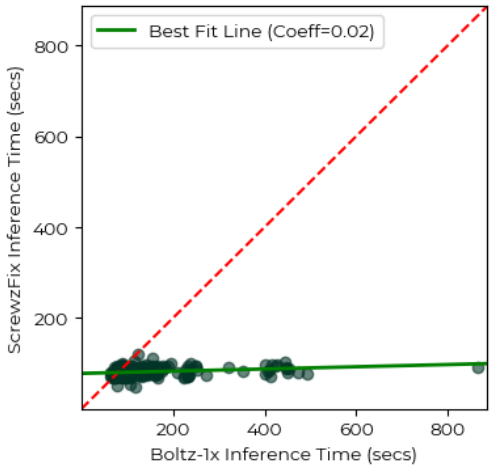


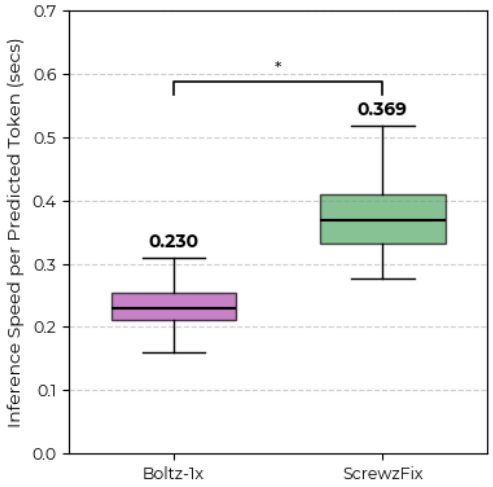
Figure 5.4: Relationship between inference time and number of tokens predicted using Boltz-1x for the Runs N Poses set. A quadratic relationship was fitted to the data, with R^2 calculated to assess the quality of the fit. Speeds measured on a NVIDIA A100 Ampere with a single CPU.

propose a simple solution by trimming the protein of regions that are not within a threshold of the pocket of interest. This strategy leverages the assumption that the distal regions of the protein are not crucial for the co-folding process of the protein-ligand complex, and therefore, removing them should not significantly impact accuracy. However, the resulting smaller, unphysiological chains fall outside the distribution for which Boltz-1 has been trained, likely resulting in increased errors. To address this, I guided the chain positions found in the original protein, reasoning that if a structure of the protein is available, whether experimentally determined or a selected co-folded structure with the conformation of interest, the trimmed prediction can always be guided to a full protein. However, despite the increase in speed by limiting the number of tokens predicted (shown in Figure

5.5a), the overhead introduced by the guidance and preprocessing for atom mapping between predicted and ground-truth structures results in slower inference speed per token (shown in Figure 5.5b).



(a) Inference speed comparison between Boltz-1x and ScrewzFix (FlexDock). The line of best fit (green) shows an almost flat relationship.



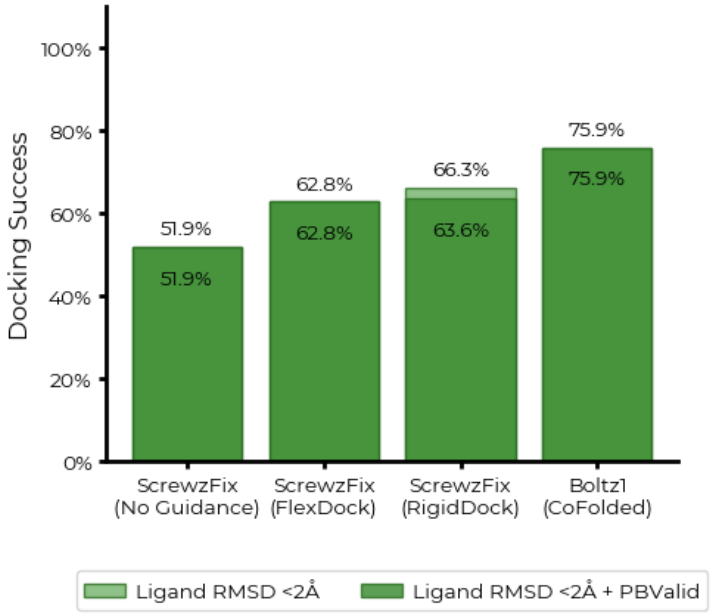
(b) Inference speed per token for Runs N Poses set between Boltz-1x and ScrewzFix (FlexDock). Median speeds are printed above the boxplots, and * denotes statistical significance at the 5% level in the difference in speeds per token between the methods. Speed measured on a NVIDIA A100 Ampere with a single CPU.

Figure 5.5: Speed analysis of Boltz-1x vs ScrewzFix: (a) overall inference speed, (b) inference speed per token. Speeds measured on NVIDIA A100 Ampere with a single CPU.

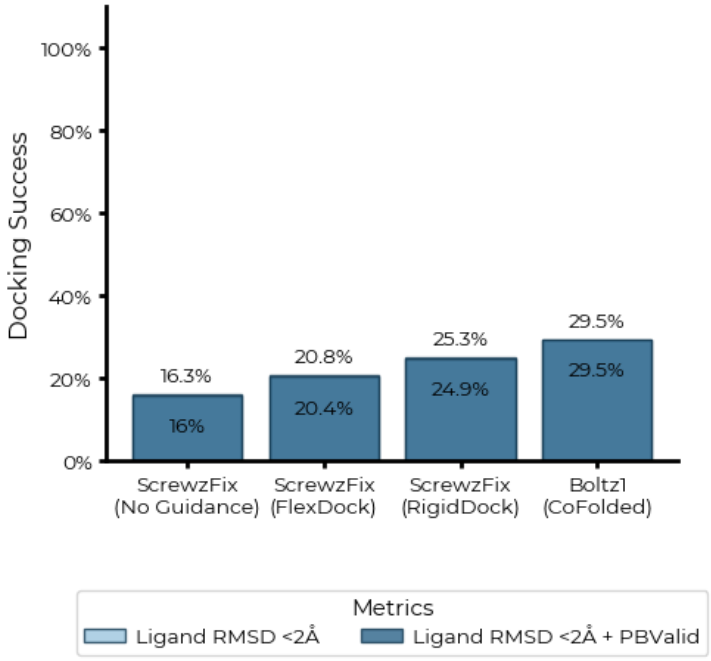
This guidance can be applied in two ways: guiding all the atoms of the protein chains to their original positions except the ligand, effectively turning the co-folding process into a rigid body docking task (ScrewzFix (RigidDock)). Alternatively,

by guiding all the atoms except the side chains within 5Å of the original ligand conformer, allowing Boltz-1 to predict these side chain orientations freely, and also the ligand, this converted the cofolding process to a flexible body docking task (ScrewzFix (FlexDock)). To measure the impact of these different guidance methods on the original cofolding process, I used the Astex and Runs N Poses sets from the previous chapter to evaluate the ligand docking accuracy of the different methods. Ligand accuracy was measured in the proposed docking classification of RMSD below 2Å and is physically valid, as measured using the PoseBusters checks (Buttenschoen et al. 2024). The accuracy was measured for one-shot prediction, so each method predicted a single example. Further analysis on the impact of multiple inferences and subsequent ranking could be explored in the future.

Maintaining the original co-folding yields the highest accuracy method, with ligand accuracy of 75.9% and 29.5% for the Astex and Runs N Poses benchmarks, respectively, as shown in Figures 5.6a and 5.6b. By just trimming the protein to the chains near the ligand-binding pocket, there is some predictive accuracy (51.9% and 16% for Astex and Runs N Poses, respectively), but this is lower than co-folding the entire protein-ligand complex. This result supports my hypothesis that predicting a complex with these "trimmed" chains results is challenging for Boltz-1x. Guidance improves the docking accuracy but does not reach the accuracy of the original co-folding. ScrewzFix (RigidDock) incrementally outperforms ScrewzFix (FlexDock) on both the Astex set (63.6% compared to 62.8%) and the Runs N Poses set (24.9% compared to 20.4%), suggesting that enabling unguided side chain predictions slightly harms docking accuracy. The gap in performance between ScrewzFix and Boltz-1x could be addressed in future work by removing the discontinuity of guiding some of the atoms and not others (such as the ligand), which might be confusing the diffusion process. However, these results show promise that this methodology can be used to replace the full co-folding of an entire protein-ligand complex without entirely ablating accuracy.



(a) Astex



(b) Runs N Poses

Figure 5.6: Ligand docking accuracy on the Astex and Runs N Poses sets for co-folding the entire protein-ligand complex with Boltz-1x (Boltz-1x), just trimming the protein and no guidance (ScrewzFix (No Guidance), guiding all atoms of the "trimmed" protein except side chains within 5Å of the ligand (ScrewzFix (FlexDock)) and guiding all atoms in the "trimmed" protein (ScrewzFix (RigidDock)). Accuracy is measured by whether the ligand is <2Å RMSD (value displayed below bar height) or both ligand is <2Å RMSD and PBValid (value displayed below bar height).

5.4.2 Ligand-conditioned side-chain packing

In the previous chapter, I demonstrated that the predictions of ligand pocket design tools produce physically implausible structures, limiting their applicability to downstream analysis. To begin to be able to analyse proposed ligand pocket sequences and their structure, the designed ligand pocket with the ligand pose would require prediction of both ligand, side chain orientations and backbone positions, which currently co-folding methods are suited to. However, it is clear from the docking accuracy of co-folding in the previous section (Figures 5.6a and 5.6b) and from other work (Škrinjar et al. 2025) that co-folding methods are not accurate enough at recapitulating the correct ligand binding pose when predicting outside of their training distribution. Therefore, using Boltz-1 or other cofolding tools to validate ligand pocket predictions that are out of distribution compared to their training data is unlikely to be helpful. However, novel ligand pockets are currently designed with a specified input ligand pose, the exception being FlowSite (Stark et al. 2023), so the complete prediction of a new ligand pose is not necessary, but instead what is needed is just an optimisation of its geometry in relation to the predicted pocket structure. This lack of prediction necessity is also true for the pocket backbone, which is also specified *a priori* before redesign. Therefore, ligand-conditioned side chain packing that allows for backbone and ligand flexibility can be used to generate useful structures of generated ligand pockets.

Here, given a known ligand pose and backbone conformation, I use my method, ScrewzFix, to predict the conformation of the pocket by also guiding the ligand atoms to the input ligand (ScrewzFix (FlexSCPack)). This method enables predictions of the side chain orientations for a given designed pocket whilst allowing small amounts of conformational flexibility for the input ligand and backbone. LigandMPNNSCPacker is an alternative method that does not alter ligand or backbone positions at all, and, as shown in the previous chapter (Chapter 4) and in this work below, produces physically implausible predictions, most likely due to this inflexibility. I compared the methods using the Runs N Poses set and utilising the accuracy threshold of 1Å RMSD for side chain accuracy and measured

physical validity using the tests curated in the previous chapter. Additionally, I also compared the ligand position accuracy as a test of the physical validity of ligand-conditioned side chain packing, using the 2Å RMSD as a threshold. If the ligand is not in the correct position after packing the pocket side chains, the generated structure is not helpful for downstream analysis, and so ligand position is an essential criterion for success when measuring performance.

Results are displayed in Table 5.2. LigandMPNNSCPacker suffers from protein-ligand clashes and side chain clashes, which cause its overall plausibility (69.8%) to be lower than that of ScrewzFix (FlexSCPack) (81.2%). ScrewzFix produces fewer protein-ligand clashes but is imperfect at producing side chains with valid conformers (Side Chain PBValidity = 87.9%) and also still produces side chain clashes (No Side Chain Clashes = 89.7%). Cofolding with Boltz-1x produces high physical plausibility, except for side chain clashes (No Side Chain Clashes = 92.1%). However, due to its poor prediction of ligand position, which the other methods keep fixed, the overall plausibility of the complex is low (27.4%). By guiding the prediction of the "trimmed" complex to that of LigandMPNNSCPacker, the physical plausibility was comparable to that of ScrewzFix (FlexSCPack) except with improved avoidance of side chain clashes (97.2%).

When examining the accuracy of the orientations of the side chain predictions alone and combined with physical plausibility, there is a clear trend shown in Figure 5.7. Boltz-1x has lower accuracy in side chain prediction (59.6%), which ScrewzFix (FlexSCPack) is close to (63.8%), suggesting co-folding is poor at predicting side chain orientations accurately, and ScrewzFix inherits this behaviour. However, as ScrewzFix enables the ligand to be fixed, the overall accuracy and validity of the generated protein-ligand complexes are higher (58% compared to 21.2%). However, LigandMPNNSCPacker is more accurate at side chain prediction (90.5%), but its poor plausibility results in an overall accuracy and validity of 65.1%. LigandMPNNSCPacker is still outperforming Boltz1 and ScrewzFix (FlexSCPack), but guiding ScrewzFix predictions to the LigandMPNNSCPacker predicted side chain orientations results in a similar accuracy (89.3%) but a higher overall accuracy

and validity (77.2%). This use of Boltz-1x to optimise an accurate but physically invalid structure could be considered a parameter-free coarse-grain minimisation of LigandMPNNSCPacker's prediction, avoiding the need for time-consuming and expensive molecular dynamics minimisation. However, the comparison to a minimisation protocol is still needed to understand the trade-off of using Boltz-1x in this way. By guiding to the cognate structure, this analysis provides an upper estimate of the accuracy of these methods; testing the accuracy of guiding to an alternate conformation or a co-folded structure is still needed to ascertain this method's usefulness in a pipeline for ligand pocket design.

Plausibility Check	**LigandMPNN-SCPacker**	**ScrewzFix (FlexSCPack)**	**Boltz-1x (Cofolded)**	**ScrewzFix + LigandMPNN**
Ligand PBValidity	96.3%	97.1%	96.9%	94.5%
Protein–Ligand PBValidity	78.0%	100.0%	99.7%	99.31%
RMSD < 2.0Å	100.0%	97.1%	29.5%	97.6%
Side Chain PBValidity	99.3%	91.3%	99.0%	87.9%
No Side Chain Clashes	89.8%	89.7%	92.1%	97.2%
Overall Plausibility	69.8%	81.2%	27.4%	81.7%

Table 5.2: Ligand-conditioned side chain packing plausibility on the Runs N Poses set for LigandMPNN side chain packer (LigandMPNNSCPacker), ScrewzFix (FlexSCPack), Boltz-1x and the combined ScrewzFix + LigandMPNNSCPacker method. The plausibility checks are detailed in the previous chapter.

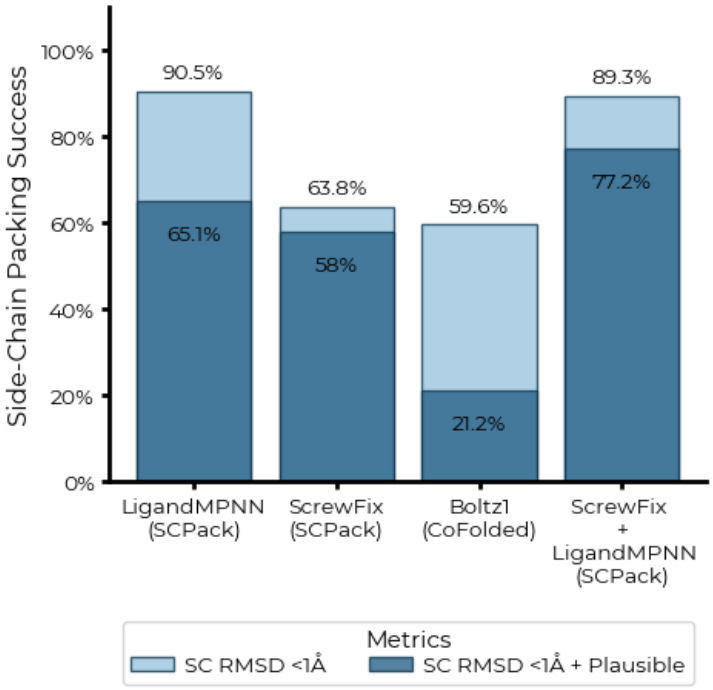


Figure 5.7: Ligand-conditioned side chain (SC) packing success for the Runs N Poses set measured by just RMSD accuracy below 1Å (SC RMSD <1Å) (values below dark blue bar height) and both RMSD accuracy below 1Å and having overall plausibility (values above light blue bar height). Methods compared are LigandMPNN side chain packer (LigandMPNNSCPacker), ScrewzFix (FlexSCPack), Boltz-1x and the combined ScrewzFix + LigandMPNNSCPacker method.

5.4.3 Preliminary results for Sparkz hallucination

Finally, by trimming down the protein and guiding its chains, structure prediction hallucination is more scalable for ligand pocket design, and as an initial prototype, I implemented a simple method that optimises the Ligand iPTM of the predicted protein-ligand complex. I chose Ligand iPTM (the predicted TMScore (Zhang et al. 2004) between interacting chains) as the confidence metric as it is accurate at discriminating accurate co-folded ligand poses from inaccurate ones (Škrinjar et al. 2025). Optimising the sequence of a pocket so that its predicted structure, when bound to a ligand, better reflects a docked pose, could provide a signal for the generation of potential binding pockets. Here, the cofolding process is initialised with the atomic structure of alanine residues for the pocket residues being redesigned, and the typical one-hot encoding of the sequence is replaced with a randomly sampled sequence probability distribution. With the backbone and ligand guided

using ScrewzFix (FlexSCPack), the Ligand iPTM prediction's gradients for the predicted complex is backpropagated through the weights of Boltz-1x and is used to optimise the sequence distribution. This process is repeated for 50 steps, described in the Data and Methods, by optimising the sequence probability compelling the distribution to tend towards a new one-hot encoded sequence. The method clearly can optimise a sequence to have higher Ligand iPTM, when repredicted using ScrewzFix, compared to randomly sampling a sequence, although it cannot reach the Ligand iPTM of the ground truth sequence (Figure 5.8).

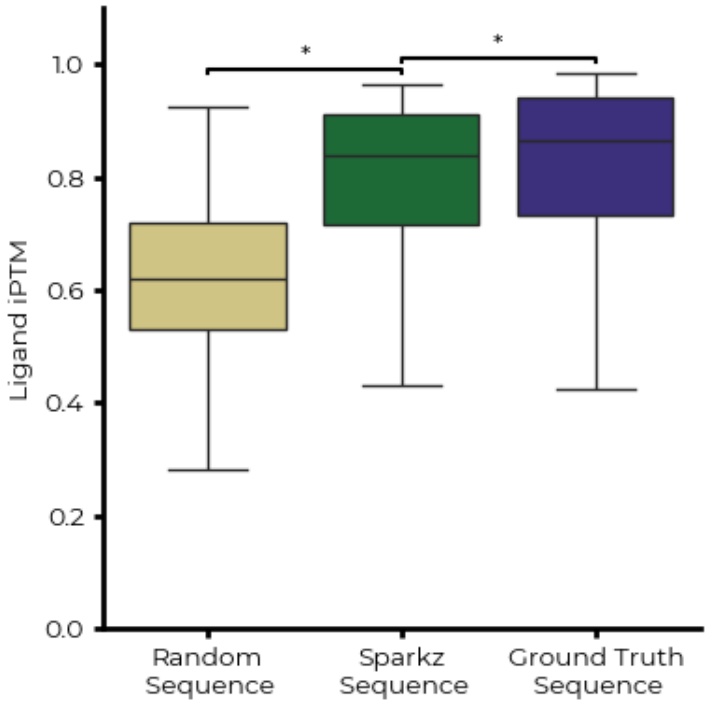


Figure 5.8: Boxplots of predicted Ligand iPTM of ligand pockets from the Runs N Poses set. Pockets are predicted using ScrewzFix (FlexSCPack) and their sequences are randomly sampled (Random Sequence), hallucinated using Sparkz (Sparkz Sequence) or the original sequence of the protein (Ground Truth Sequence). Significantly different (<5%) between distributions, calculated using the Mann-Whitney U test, are denoted with *.

However, Sparkz does not appear to explore sequence space to generate binding sequences when examined using the MET kinase benchmark developed in the previous chapter, Other methods are only capable of recapitulating the ground truth sequence; however, by optimising Ligand iPTM, the same trend occurs to a

lesser extent, as shown in Figure 5.9. For all inhibitors except Tiviantinib, Sparkz enriches for WT mutations significantly more than random, but does not do the same for any other type of mutation class. Therefore, it seems simply optimising Ligand iPTM is not a useful strategy for generating binding ligand pocket sequences. However, as it is simple to change the confidence metrics or loss to optimise for in Sparkz, exploring which predicted confidence or combined confidence might show better results on this benchmark can be explored. Furthermore, integrating the outputs of other models, such as protein large language models, as part of the optimisation may also prove fruitful. However, it is essential to note that this result is for a single protein, and development should be avoided to optimise metrics against this benchmark solely. This inadequacy highlights the need for curation of other deep mutational scan datasets similar to the one developed in this study (Estevam et al. 2024).

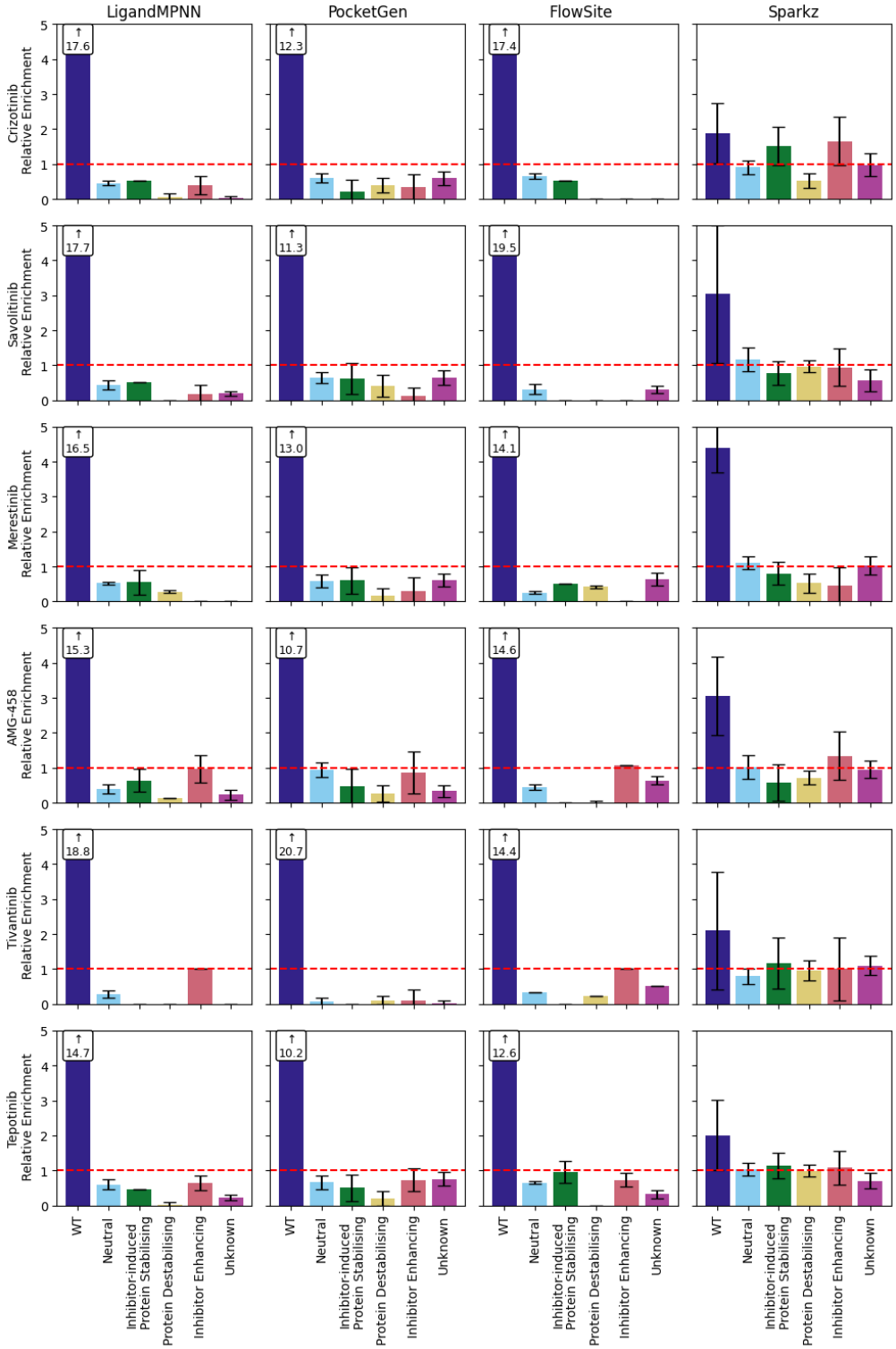


Figure 5.9: Relative enrichments of mutation classes, classified based on deep mutational scanning data from (Estevam et al. 2024) for the MET Kinase for different ligand-pocket generation methods (LigandMPNN, PocketGen, FlowSite and Sparkz). An enrichment of 1, which is the probability of a class of mutation being picked relative to the probability of randomly picking that mutation, is indicated as a red dashed line.

5.5 Discussion

This preliminary work aims to both enable faster prediction of generated ligand-binding pockets with higher accuracy and physical plausibility, and to facilitate the generation of sequences for ligand-binding pockets. This faster inference, implemented for the Boltz-1x cofolding method, is achieved by reducing the number of protein residues needed for prediction to only co-folding residues close to the ligand-binding pocket, thereby "trimming" the protein. By guiding these chains to a known whole protein, I developed the ScrewzFix method. This approach partially corrects for the error introduced by predicting these unfamiliar shortened chains, but further improvements are needed.

For the task of predicting the side-chain orientations of a sequence for a redesigned ligand pocket, ligand-conditioned side-chain packing, ScrewzFix, outperforms Boltz-1, as it prevents the incorrect prediction of ligand pose by guiding it to the specified position. However, when combined with the more accurate but less physically valid generations of LigandMPNNSCPacker, by guiding Boltz-1x to predict LigandMPNNSCPacker's predicted side chain orientations, physical plausibility is improved, producing more accurate and valid structures than ScrewFix and LigandMPNNSCPacker alone. Further analysis is needed to determine if these differences in performance hold true when using a predicted or non-cognate crystal structure as the starting protein structure.

Finally, by using this faster inference, I was able to implement a novel structure prediction-based hallucination method, called Sparkz. Preliminary results indicate that it is capable of generating ligand-pocket sequences with higher Ligand iPTMs than random sequences. However, using deep mutational scanning data, it was found that this first version of the method, albeit to a lesser extent than other tested ligand pocket generation methods, recapitulates the original pocket sequence. Both of these developments show promise in addressing the problems raised in Chapter 4 in the field of ligand pocket generation, and so enabling the generation of synthetic protein-ligand complexes for training other models, but they require

further work. In the next chapter, I summarise the conclusions of all the results chapters and discuss future directions for my research.

6 Conclusions

The combination of SBDD and ML has promised to drive forward progress in the field of small-molecule drug discovery. By learning from existing structure experimentally determined by researchers all over the world, ML algorithms could learn a generalisable understanding of where a small molecule binds to a protein, how well it binds and what possible molecules could bind to a given pocket. This thesis has focused on how structural data influences the performance and interpretability of these methods, and introduces pipelines and methodologies to improve both the data and the models trained on it.

6.1 Binding Affinity Prediction

The prediction of binding affinity for any protein-ligand complex by training ML models, MLBSFs, on static structures is still a popular problem in the field. In Chapter 2, I examined whether diverse methods are learning the underlying physics of binding or memorising trends and similarities in the training data when trained on static, labelled protein-ligand complexes from PDBBind (Wang et al. 2005), a popular dataset. By comparing to baseline models that I designed to be obfuscated from learning the underlying physics through, I found that the methods matched

the performance of these baselines, suggesting they too were learning not much more than these dataset biases.

In the chapter, one key benchmark developed was the 0 Ligand Bias set, which used identical ligands binding to different proteins with different binding affinities to penalise learning these biases at test time. With the development of cofolding (Abramson et al. 2024; Wohlwend et al. 2025) after the completion of this work, it is now possible to dock ligands and proteins with no need for a structure. By accurately docking structures of identical ligands bound to different proteins from databases with no associated structure, like ChEMBL (Gaulton et al. 2012), it could be possible to develop much larger training sets that penalise learning biases during training. This could provide a path towards overcoming the limitations highlighted in Chapter 2.

6.2 Pose Classification

Pose classification can be a vital part of a docking software or pipeline as it enables the triaging of docking outputs and so better interpretation of its prediction. In Chapter 3, I explored the impact of noising accurate poses, using the developed PoseFoundry pipeline, to be out-of-distribution by pose classifiers trained on the outputs of a specific docking software, Smina (Koes et al. 2013). I found that this artificial and adversarial noise did harm performance, and by training on this noise, I was able to recapture accuracy again. However, learning from these artificial poses did not result in improved accuracy compared to training on Smina poses, except for scoring poses from DiffDock (Corso et al. 2022) into cofolded apo structures.

Despite these disappointing results, there are some promising future directions for the development of this work. One avenue to explore would be the application of pose classifiers trained on Augmented2020, the dataset produced by PoseFoundry, as guidance functions for diffusion-based docking methods such as DiffDock and Boltz (Wohlwend et al. 2025). Classifier-based guidance struggles if the classifier is not robust to noise, but in this chapter, I specifically trained for this robustness. Whether using such guidance could improve the accuracy of these methods would

be an interesting extension of this chapter. A limitation of this chapter is the use of a relatively simple EGNN architecture that did not explicitly encode hydrogens, partial charges or waters. By increasing the complexity and size of the model, performance gains could be found.

6.3 Synthetic protein-ligand structures

Chapters 4 and 5 set out the start of work to produce synthetic protein-ligand complex structures that could be used to expand and balance the existing experimental data curated in the PDB. In Chapter 4, I explored the capabilities of ligand-pocket generation methods that could predict sequences for a given ligand pose and protein scaffold, and so predict diverse proteins that could bind the same ligand. As part of this prediction, they also predict structures of the protein-ligand complex. First, I tested the physical plausibility of these generated structures using curated tests and found that they frequently physically implausible, generating protein-ligand and side chain-side chain clashes. Furthermore, I showed that implausibility inflates the prediction of binding energy by the AutoDock Vina scoring function, a metric used to benchmark these methods. I developed tests and benchmarks to more rigorously understand whether they have learnt underlying ligand-pocket protein sequence relationships using adversarial tests, and I curated a deep mutation scanning benchmark for the MET kinase (Estevam et al. 2024) By trying to teach these methods ligand-pocket protein sequence relationships by recapitulating the ground truth sequences, these methods did not sufficiently explore sequence space.

Building on these observations, Chapter 5 focuses on developing a faster co-folding approach, based on Boltz-1x, to enable rapid inference and systematic analysis of generated ligand pockets, a method called ScrewzFix. By trimming the protein into smaller chains and guiding these unphysiological chains, I was able to get close to the accuracy of co-folding the entire protein-ligand complex, but by predicting fewer residues. This faster inference also enabled the development of an initial structure-prediction hallucination method for ligand pocket design, Sparkz.

Although Sparkz successfully optimised the confidence metric, it did not appear to optimise for mutations that favour binding, according to the MET kinase benchmark.

This work offers many avenues for future work. Firstly, the gap in performance between ScrewzFix and Boltz-1x could be addressed by improving the atom guidance. Currently, atoms that are freely predicted are not guided at all, resulting in a discontinuity between them and the guided atoms, which could harm accuracy. To address this, the unguided atoms could also be guided based on the average vectors of the nearest guided atoms to them at each time step. Next, Sparkz could be built upon with more complex loss terms to optimise, such as the pLDDT or using other models to optimise. Implementing the method to work on the new Boltz-2 model would also enable optimising according to its binding affinity predictions as well as leveraging its faster inference speeds (Passaro et al. 2025). ESM IF1 (Hsu et al. 2022) is an inverse-folding methodology that has demonstrated high accuracy in predicting mutational effects on proteins (Notin et al. 2023). Optimising the sequence according to its output logits could help avoid predicting residues that destabilise the protein and limit the search space for residues that improve ligand binding. However, this work is limited by being benchmarked against a single protein, the MET kinase. The curation of further deep mutational scan datasets that explicitly measure the impact of mutations on binding will be vital for validating any method developments.

A different approach for ligand pocket design to pursue is to adopt constraints on prediction based on interactions present in a structure. This approach has been successfully applied to generating ligands for a given pocket and set of interactions, such as MolSnapper (Ziv et al. 2025) and SILVR (Runcie et al. 2023). However, this could be inverted to create protein pocket sequences that satisfy specific interactions with the ligand by using recently developed de novo molecule generation methods. One promising example is LaProteina (Geffner et al. 2025), a method that co-predicts atomistic protein structure and sequence using latent space flow matching for side chain conformations.

Finally, this work aims to develop a pipeline to generate these synthetic protein-ligand complexes eventually. By generating ligand pockets, filtering and analysing them with techniques such as molecular dynamics and experimental assays, it may be possible to expand upon the structures of the PDB without having to explicitly determine them experimentally.

6.4 Closing Remarks

This thesis has highlighted the central role of structural data in shaping the performance and reliability of ML methods for structure-based drug discovery. A recurring theme in the first chapters has been the interrogation of models that appear accurate but in reality exploit dataset biases and so do not generalise out of distribution. Addressing these biases by designing better benchmarks, generating synthetic structural data, and developing pipelines that probe what models are actually learning will be essential if ML is to deliver robust, generalisable insights for drug discovery.

Looking forward, expanding the structural landscape beyond experimentally determined complexes will be key. Initiatives such as OpenBind (OpenBind 2025) and OpenADMET (MacDermott-Opeskin et al. 2025) offer the possibility of larger, more diverse, and less biased training resources and could complement my synthetic protein-ligand data. However, ensuring that these new datasets do not simply reproduce the limitations of existing resources will require careful curation, benchmarking, and validation. Ultimately, the progress of this field will depend not only on advances in algorithms but also on advances in the structural data itself: how it is collected, curated, and augmented. By interrogating existing biases, building principled methods to expand structural datasets, and developing tools to assess model trust, we can move toward a future where ML for SBDD does not just mimic existing knowledge but actively drives the discovery of new therapeutics.

Appendices

A

Robustly interrogating machine learning-based scoring functions: what are they learning?

A.1 Datasets

A.1.1 PDB IDs of Holo CASF 2016 proteins and their respective Apo PDB ID

Holo	Apo
2xb8 3n76 3n7a 3n86 4ciw	2dhq
1nc1 1nc3 1y6r 4f2w 4f3c	1z5p
3u8k 3u8n 3zdg 4qac 3wtj	3sq9
1e66 1gpk 1gpn 1h22 1h23	7b38
2wvt 2xii 4j28 4jfs 4pcs	4j27
1ps3 3d4z 3dx1 3dx2 3ejr	3bub
1z95 3b5r 3b65 3b68 3g0w	None
3qqs 3r88 3twp 4gkm 4owm	3qr9
2fxs 2iwx 2vw5 2wer 2yge	1ah6
2cbv 2cet 2j78 2j7h 2wbg	1od0
2r9w 3gr2 3gv9 4jxs 4kz6	6t3d
3g2z 3g31 4de1 4de2 4de3	2p74
3nq9 3ueu 3uev 3uew 3uex	1b8e
2vkm 3rsx 3udh 4djv 4gid	2zhv
1k1i 1o3f 1uto 3gy4 4abg	5mnz
3p5o 3u5j 4lzs 4ogj 4wiv	4lyi
3ui7 3uuo 4llx 5c28 5c2h	2oup
1q8t 1q8u 1ydr 1ydt 3ag9	None
2weg 3dd0 3kwa 3ryj 4jsz	3ks3
3nw9 3oe4 3oe5 3ozs 3ozt	4pym
1pxn 2fvd 2xnb 3pxf 4eor	4ek3
4agn 4agp 4agq 5a7b 5aba	6shz
3arp 3arq 3aru 3arv 3ary	3b8s
1lpg 1mq6 1z6e 2xbv 2y5h	1hcg
4cr9 4cra 4crc 4ty7 4x6p	None
2zcq 2zcr 2zy1 3acw 4ea2	2zco
2v00 3prs 3pww 3uri 3wz8	5rdh
1qkt 2p15 2pog 2qe4 4mgd	None

Holo	Apo
1p1n 1p1q 1syi 2al5 4u4s	1fto
1vso 3fv1 3fv2 3gbb 4dld	None
3ebp 3g2n 3l7b 3syr 4eky	3e3l
1yc1 2xdl 2yki 3b27 3rlr	5j2v
3ao4 3zso 3zsx 3zt2 4cig	None
3ehy 3lka 3nx7 3tsk 4gr0	1os9
2zb1 3e92 3e93 4dli 4f9w	4e5b
2vvn 2w4x 2w66 2wca 2xj7	4ais
3coy 3coz 3ivg 4ddh 4ddk	3cov
2p4y 2yfe 3b1m 3fur 3u9q	6l8b
1a30 1eby 2qnq 3o9i 1g2k	3phv
1r5y 1s38 3gc5 3ge7 3rr4	4pun
1o0h 1u1b 1w4o 3d6q 3dxg	6etk
3cj4 3gnw 4eo8 4ih5 4ih7	1nb4
2wtv 3e5a 3myg 3uo4 3up2	6cpe
1nvq 2br1 2brb 3jvr 3jvs	1ia8
2c3i 3bgz 3jya 4k18 5dwr	1xqz
2wn9 2wnc 2x00 2xys 2ymd	2byn
3kr8 4j21 4j3l 4kzq 4kzu	3kr7
1qf1 1z9g 3fcq 4tmn 5tmn	2g4z
1bcu 1oyt 2zda 3bv9 3utu	4nzq
4bkt 4w9c 4w9h 4w9i 4w9l	3zrf
3f3a 3f3c 3f3d 3f3e 4mme	5jae
2v7a 3k5v 3mss 3pyy 4twp	None
3qgy 4m0y 4m0z 4qd6 4rfm	None
4e5w 4ivb 4ivc 4ivd 4k77	None
4e6q 4f09 4gfm 4hge 4jia	None
1bzc 2hb1 2qbp 2qbq 2qbr	5k9v
1c5z 1o5b 1owh 1sqa 3kgp	4dw2

A.2 Models

A.2.1 Model Implementation Differences

MLBSF	Modification
RFScore Ballester et al. 2010	I reimplemented the model with RandomForestRegressor, using the original hyperparameters, from the SciKit package Pedregosa et al. 2011 with features taken from the Open Drug Discovery Toolkit (ODDT) Wojcikowski et al. 2015 package. Although the hyperparameters could not be perfectly matched due to differences in the Random Forest implementations, it is unlikely that this significantly affected performance.
PointVS Scantlebury et al. 2023	The model's training required a validation set for early stopping; I used a random sample of 1000 data points from the training set.
Pafnucy Stepniewska-Dziubinska et al. 2018	The original implementation used partial charges assigned by ChimeraX Goddard et al. 2018; however, due to its licensing complexity as a dependency, all charges werre set to zero.
SIGN Li et al. 2021	Like Pafnucy, SIGN uses partial charge features for the protein pocket from ChimeraX, so I set these features to zero. Its training also required a validation set for early stopping; a random sample of 1000 data points from the training set was used.
OnionNet-2 Wang et al. 2021b	In the original implementation, they used a loss function that combined Pearson's R and RMSE, PCC-RMSE. However, I found that this loss function often produced NaN losses during training, so the RMSE loss function provided in their codebase was used instead. Its training also required a validation set for early stopping; a random sample of 1000 data points from the training set was used.

A.2.2 Impact of protein pocket distance cutoff on accuracy for baseline models on CASF 2016 benchmark

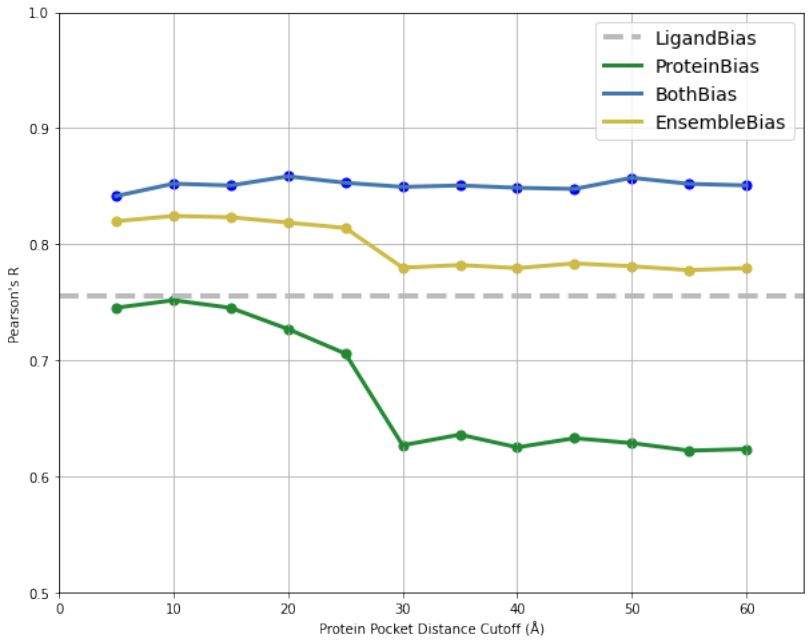


Figure A.1: Relationship between Pearson's R of baseline models on the CASF 2016 benchmark and the protein pocket distance cutoff, from any protein atom to any ligand atom, to include residues as being part of the protein pocket for featurisation.

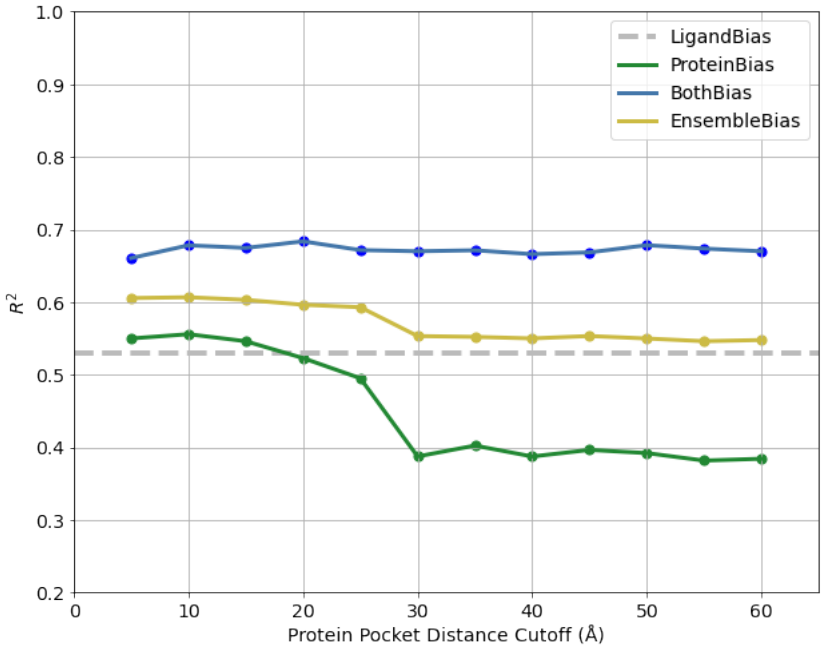


Figure A.2: Relationship between R^2 of baseline models on the CASF 2016 benchmark and the protein pocket distance cutoff, from any protein atom to any ligand atom, to include residues as being part of the protein pocket for featurisation.

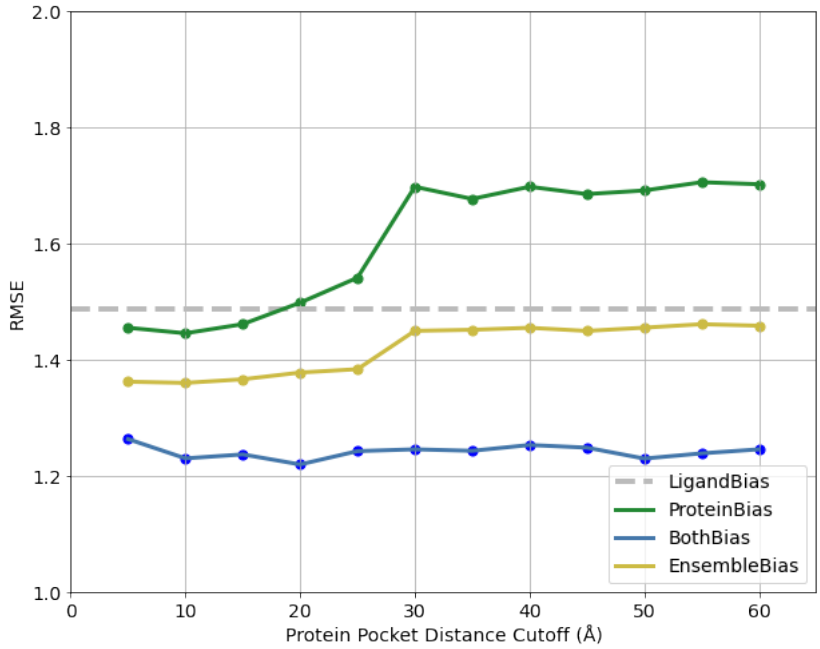


Figure A.3: Relationship between RMSE of baseline models on the CASF 2016 benchmark and the protein pocket distance cutoff, from any protein atom to any ligand atom, to include residues as being part of the protein pocket for featurisation.

A.2.3 Hyperparameters of baseline models

LigandBias		**ProteinBias**		**BothBias**	
Random Forest Regressor		Random Forest Regressor		LGBMRegressor	
n_estimators	362	*n_estimators*	1447	*n_estimators*	205
max_features	0.51	*max_features*	0.32	*num_leaves*	291
max_leaf_nodes	2038	*max_leaf_nodes*	5460	*min_child_samples*	2
				learning_rate	0.03
				log_max_bin	9
				colsample_bytree	0.6
				reg_alpha	0.01
				reg_lambda	0.01

A.3 Crystal Structure Benchmark Results

A.3.1 Further metrics for CASF 2016, 2019 Holdout, Peptides Holdout and 0 Ligand Bias

Method	CASF 2016		2019 Holdout		Peptides Holdout		0 Ligand Bias	
	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE
LigandBias	$0.53_{\pm.07}$	$1.49_{\pm.12}$	$0.33_{\pm.04}$	$1.45_{\pm.05}$	$-0.37_{\pm.08}$	$1.81_{\pm.04}$	$-0.45_{\pm.24}$	$1.93_{\pm.14}$
ProteinBias	$0.55_{\pm.09}$	$1.46_{\pm.14}$	$0.34_{\pm.04}$	$1.44_{\pm.05}$	**$0.06_{\pm.04}$**	**$1.50_{\pm.05}$**	**$0.11_{\pm.11}$**	**$1.51_{\pm.14}$**
EnsembleBias	$0.60_{\pm.06}$	$1.37_{\pm.11}$	$0.44_{\pm.03}$	$1.33_{\pm.04}$	$0.01_{\pm.04}$	$1.54_{\pm.04}$	$-0.03_{\pm.13}$	$1.62_{\pm.12}$
BothBias	**$0.67_{\pm.05}$**	**$1.24_{\pm.10}$**	$0.45_{\pm.04}$	$1.32_{\pm.05}$	$-0.14_{\pm.06}$	$1.65_{\pm.04}$	$-0.10_{\pm.16}$	$1.68_{\pm.13}$
Smina	$0.27_{\pm.13}$	$1.86_{\pm.16}$	$-0.95_{\pm.23}$	$2.49_{\pm.11}$	$-2.31_{\pm.30}$	$2.82_{\pm.10}$	$-2.31_{\pm.81}$	$2.91_{\pm.24}$
RFScore	$0.60_{\pm.05}$	$1.37_{\pm.11}$	$0.41_{\pm.04}$	$1.36_{\pm.05}$	$-0.26_{\pm.07}$	$1.73_{\pm.04}$	$-0.13_{\pm.15}$	$1.70_{\pm.13}$
PointVS	$0.58_{\pm.06}$	$1.40_{\pm.10}$	$0.44_{\pm.04}$	$1.33_{\pm.04}$	$-0.03_{\pm.06}$	$1.57_{\pm.04}$	$-0.06_{\pm.15}$	$1.65_{\pm.13}$
Pafnucy	$0.51_{\pm.06}$	$1.51_{\pm.11}$	$0.35_{\pm.05}$	$1.44_{\pm.06}$	$-0.30_{\pm.08}$	$1.77_{\pm.04}$	$-0.16_{\pm.15}$	$1.73_{\pm.12}$
SIGN	**$0.67_{\pm.07}$**	**$1.24_{\pm.11}$**	$0.41_{\pm.05}$	$1.36_{\pm.05}$	$-0.03_{\pm.07}$	$1.57_{\pm.05}$	$-0.17_{\pm.18}$	$1.73_{\pm.14}$
OnionNet-2	$0.66_{\pm.06}$	$1.26_{\pm.10}$	**$0.49_{\pm.04}$**	**$1.28_{\pm.05}$**	$0.05_{\pm.05}$	$1.51_{\pm.04}$	$-0.04_{\pm.16}$	$1.64_{\pm.15}$

Table A.3: R^2 and RMSE (in pK units) between predicted and true pK values for protein-ligand complexes for the baseline models (LigandBias, ProteinBias, EnsembleBias and BothBias), a non-ML-based scoring function (Smina) and five commonly used MLBSFs (RFScore, PointVS, Pafnucy, SIGN and OnionNet-2) on four benchmark datasets (CASF 2016, 2019 Holdout, Peptides Holdout and 0 Ligand Bias). See methods for further details of scoring functions and dataset creation. Error ranges represent the 95% confidence intervals from bootstrapped R^2 and RMSE (N=10000). The highest values are in bold and underlined, with any value within the highest values' confidence intervals underlined.

A.3.2 Peptides Holdout Analysis

To examine whether low performance on Peptides Holdout is due to their low drug-likeness Bickerton et al. 2012, I restricted the set to any peptide with 10 amino acids or fewer, as 80% of oral peptide drugs were found to be within this range Santos et al. 2016. The distribution of the peptide lengths of the Peptide Holdout ligands is shown in Figure A.4 and it shows that this roughly halves the available peptides. Of the 2573 complexes, 1228 are below this threshold.

Performance of all scoring functions did incrementally increase with this subset, as shown in Table A.4, demonstrating that the longer peptides were harder to score. SIGN is now the best performing method, but is within confidence intervals of EnsembleBias, ProteinBias and OnionNet-2 across the different metrics. However,

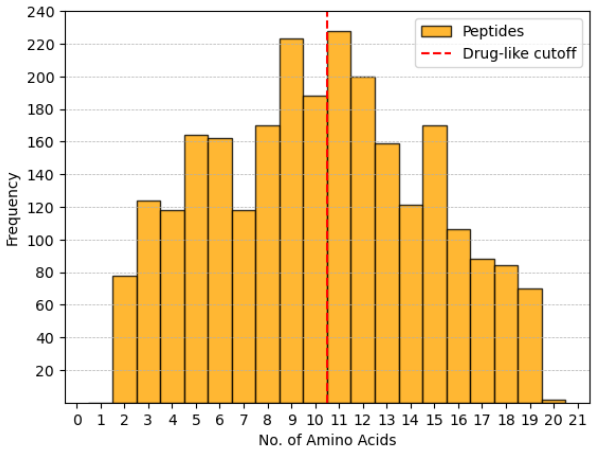


Figure A.4: Histogram of the peptide amino acid lengths for the Peptide Holdout test set. The red line indicates the 10 amino acid threshold chosen from Santos et al. 2016.

performance on the whole is still very low, showing that they still have not learnt the biophysics necessary to score these peptide-protein complexes.

Method	Peptides Holdout <10 amino acids		
	r	R^2	RMSE
LigandBias	0.31±.05	-0.19±.10	1.83±.06
ProteinBias	0.42±.05	0.15±.06	1.54±.07
EnsembleBias	0.46±.05	0.11±.06	1.58±.06
BothBias	0.43±.05	0.02±.08	1.66±.06
Smina	0.20±.05	-1.60±.33	2.70±.15
RFScore	0.40±.05	-0.02±.08	1.69±.06
PointVS	0.44±.05	0.09±.08	1.59±.06
Pafnucy	0.43±.05	-0.11±.09	1.76±.06
SIGN	**0.49±.04**	**0.18±.07**	**1.51±.06**
OnionNet-2	0.46±.05	0.13±.07	1.56±.06

Table A.4: Pearson's R (r), R^2 and RMSE (in pK units) between predicted and true pK values for protein-ligand complexes for the baseline models (LigandBias, ProteinBias, EnsembleBias and BothBias), a non-ML-based scoring function (Smina) and five commonly used MLBSFs (RFScore, PointVS, Pafnucy, SIGN and OnionNet-2) on Peptides Holdout crystal structures when restricted to peptides of length 10 amino acids or less. Error ranges represent the 95% confidence intervals from bootstrapped Pearson's R, R^2 and RMSE (N=10000). The highest values are in bold and underlined, with any value within the highest values' confidence intervals underlined.

A.3.3 Molecular Weight Correlations

Dataset	Pearson's R
CASF 2016	0.503 ± 0.084
2019 Holdout	0.195 ± 0.048
0 Ligand Bias	0.021 ± 0.109
Peptides Holdout	0.162 ± 0.041

Table A.5: Pearson correlation between ligand molecular weight (MW) and true pK values for different benchmarks. Error ranges represent the 95% confidence intervals from bootstrapped Pearson's R (N=10000).

A.4 Further metrics for scoring functions and baselines models on protein family hold-out tests

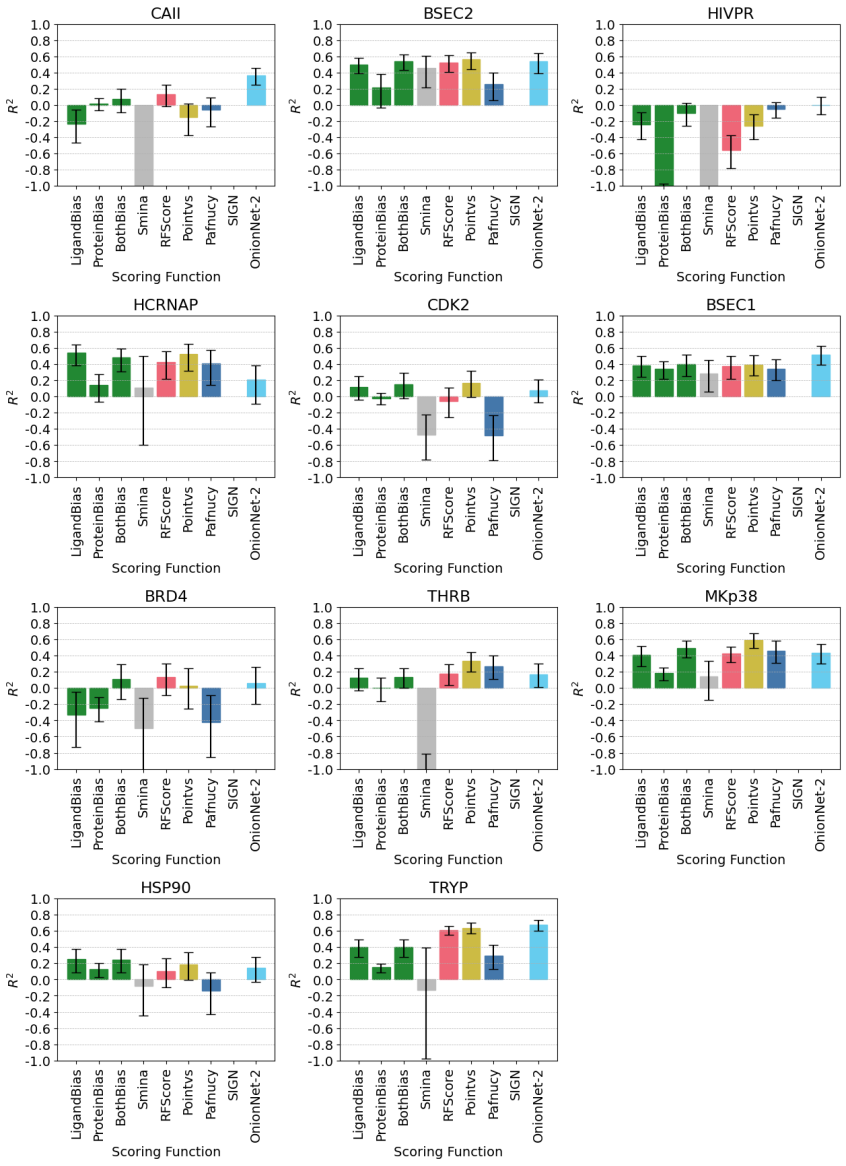


Figure A.5: R^2 between predicted and true pK values for protein-ligand complexes for the baseline models (Ligand Bias, Protein Bias and Both Bias), a non-ML-based scoring function (Smina) and five commonly used MLBSFs (RFScore, PointVS, Pafnucy, SIGN and OnionNet-2) for eleven protein family hold-out clusters. These eleven families are Carbonic Anhydrase II (CAII), Beta-secretase (BSEC2), HIV protease (HIVPR), Hepatitis C Virus RNA-polymerase (HCRNAP), Cyclin-dependent kinase 2 (CKD2), Beta-secretase (BSEC1), Bromodomain-containing protein 4 (BRD4), Thrombin (THRB), MAP Kinase p28 (MKp38), Heat Shock Protein 90 (HSP90) and Trypsin (TRYP). Error bars represent the 95% confidence intervals from bootstrapped R^2 (N=10000).

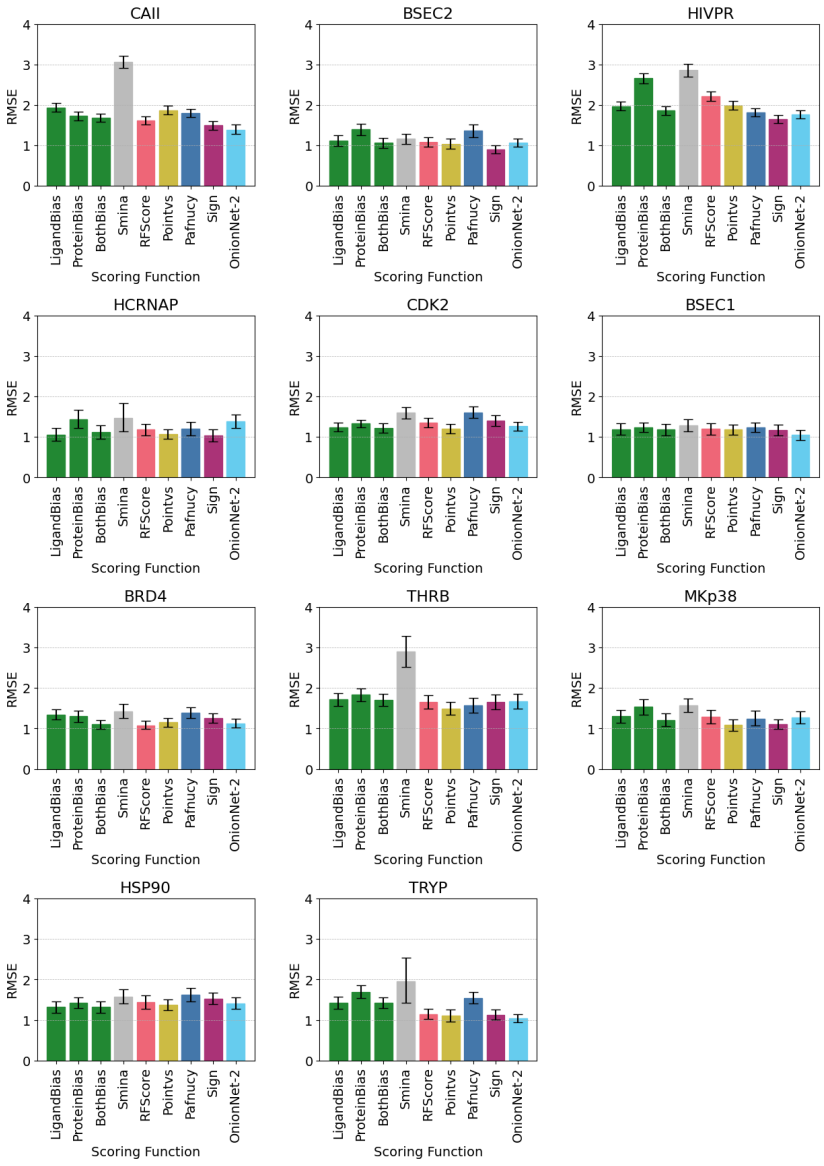


Figure A.6: RMSE between predicted and true pK values for protein-ligand complexes for the baseline models (Ligand Bias, Protein Bias and Both Bias), a non-ML-based scoring function (Smina) and five commonly used MLBSFs (RFScore, PointVS, Pafnucy, SIGN and OnionNet-2) for eleven protein family hold-out clusters. These eleven families are Carbonic Anhydrase II (CAII), Beta-secretase (BSEC2), HIV protease (HIVPR), Hepatitis C Virus RNA-polymerase (HCRNAP), Cyclin-dependent kinase 2 (CKD2), Beta-secretase (BSEC1), Bromodomain-containing protein 4 (BRD4), Thrombin (THRB), MAP Kinase p28 (MKp38), Heat Shock Protein 90 (HSP90) and Trypsin (TRYP). Error bars represent the 95% confidence intervals from bootstrapped RMSE (N=10000).

A.5 Further metrics for scoring functions and baselines models on different complex types of CASF 2016

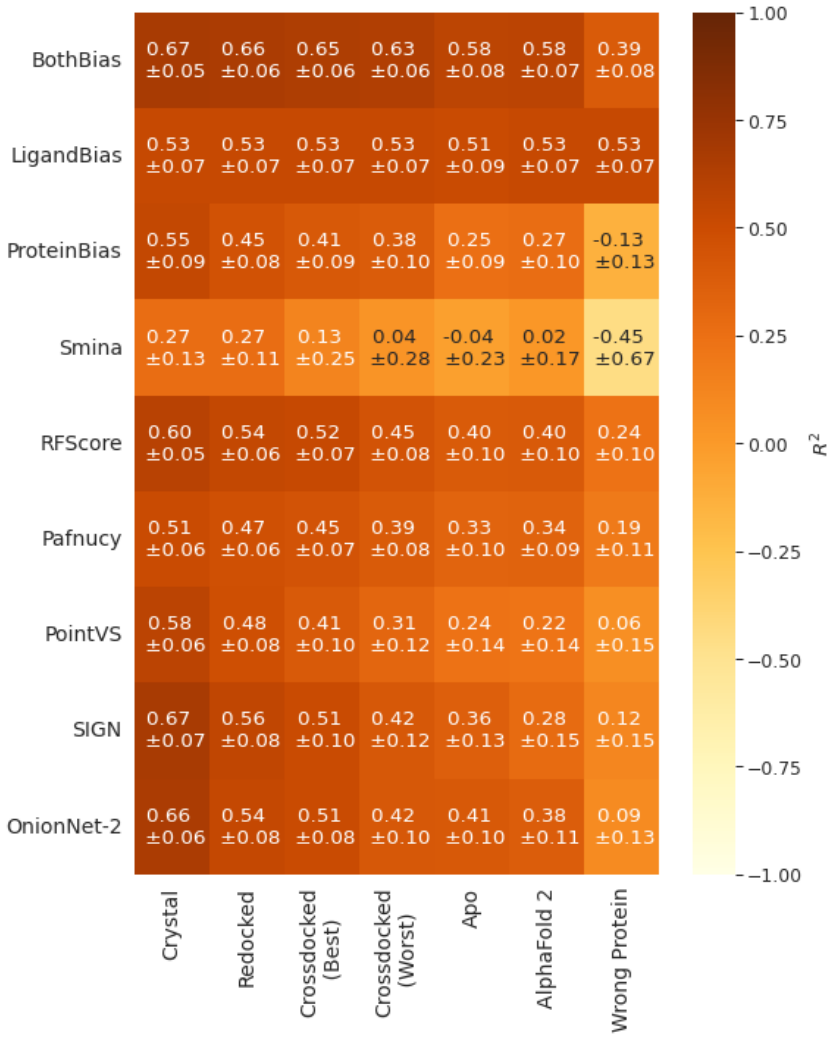


Figure A.7: R^2 between predicted and true pK values for protein-ligand complexes for the baseline models (LigandBias, ProteinBias and BothBias), a non-ML-based scoring function (Smina) and five commonly used MLBSFs (RFScore, PointVS, Pafnucy, SIGN and OnionNet-2) on alternate CASF 2016 complex type test sets. Errors are the 95% confidence intervals from the bootstrapped R^2

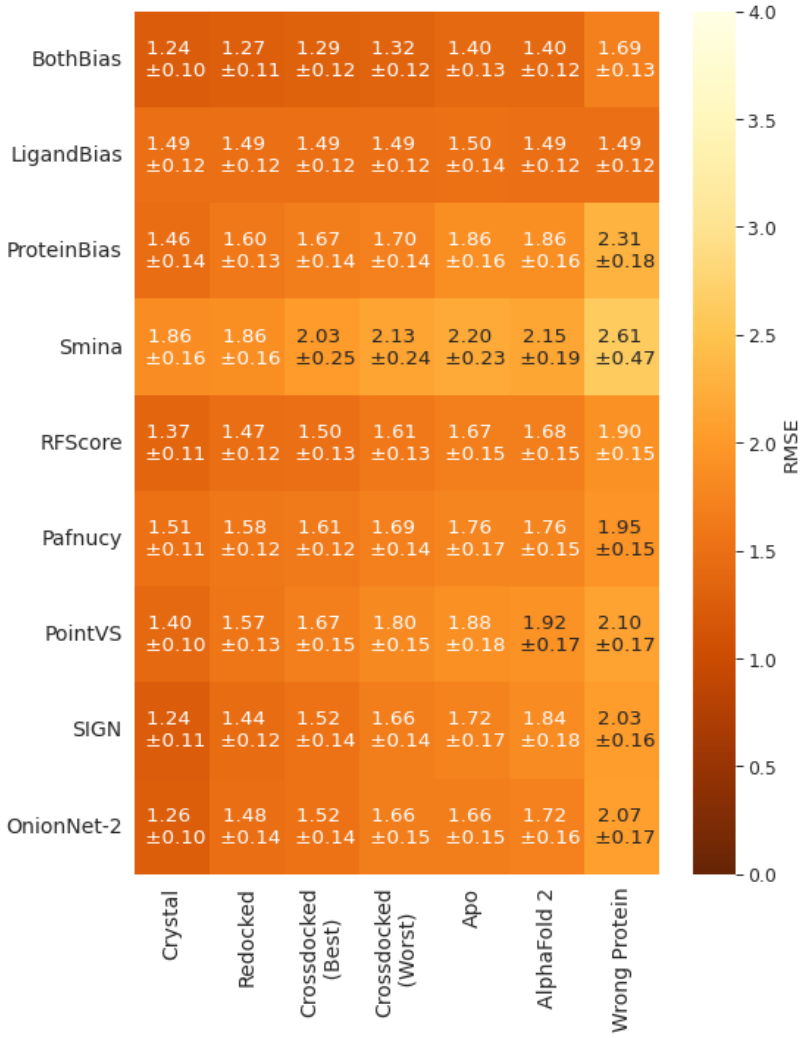


Figure A.8: RMSE between predicted and true pK values for protein-ligand complexes for the baseline models (LigandBias, ProteinBias and BothBias), a non-ML-based scoring function (Smina) and five commonly used MLBSFs (RFScore, PointVS, Pafnucy, SIGN and OnionNet-2) on alternate CASF 2016 complex type test sets. Errors are the 95% confidence intervals from the bootstrapped RMSE.

A.6 Further metrics for scoring functions and baseline models on differing docking accuracy versions of CASF 2016

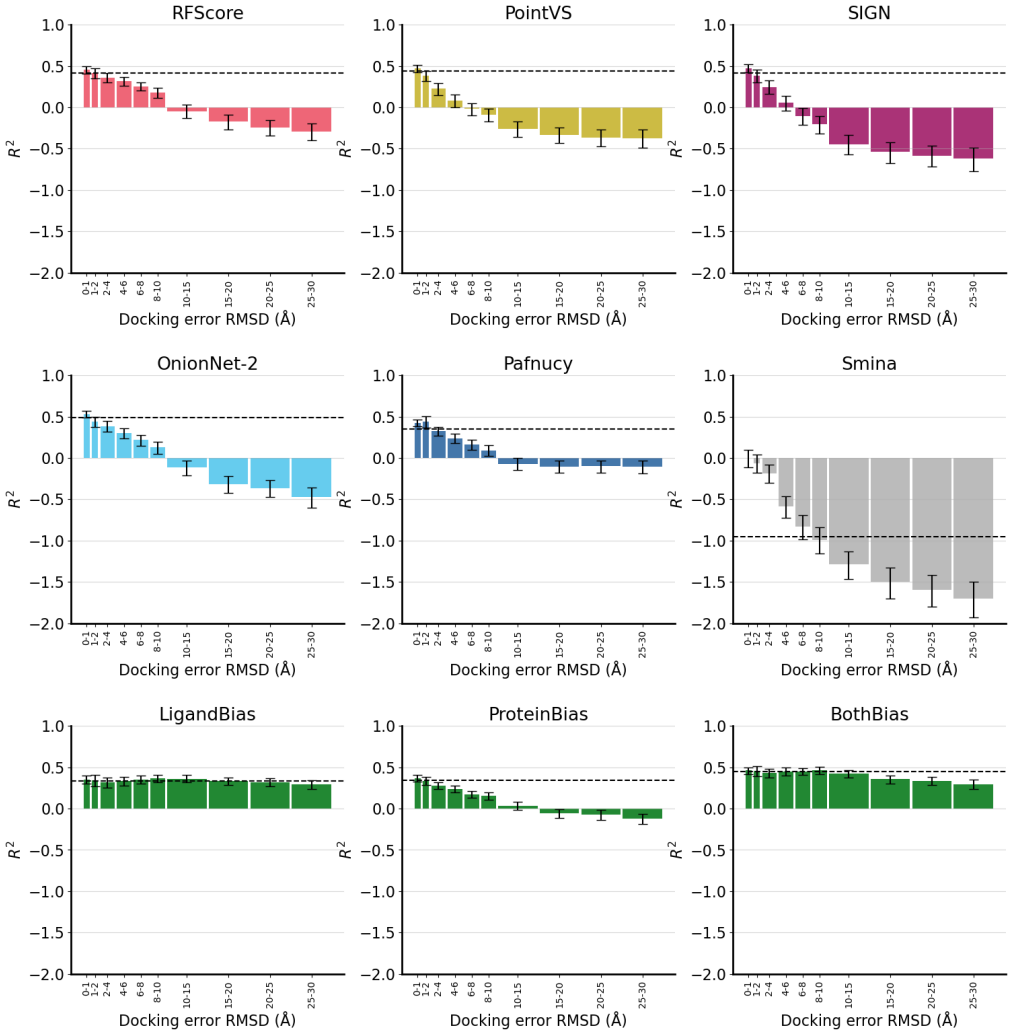


Figure A.9: R^2 between predicted and true pK values for protein-ligand complexes for the baseline models (LigandBias, ProteinBias and BothBias), a non-ML-based scoring function (Smina) and five commonly used MLBSFs (RFScore, PointVS, Pafnucy, SIGN and OnionNet-2), on different accuracy poses of CASF 2016 complexes. Accuracy on the crystal structures of CASF 2016 is shown as a dashed black line. Errors are the 95% confidence intervals from the bootstrapped R^2 (N=10000).

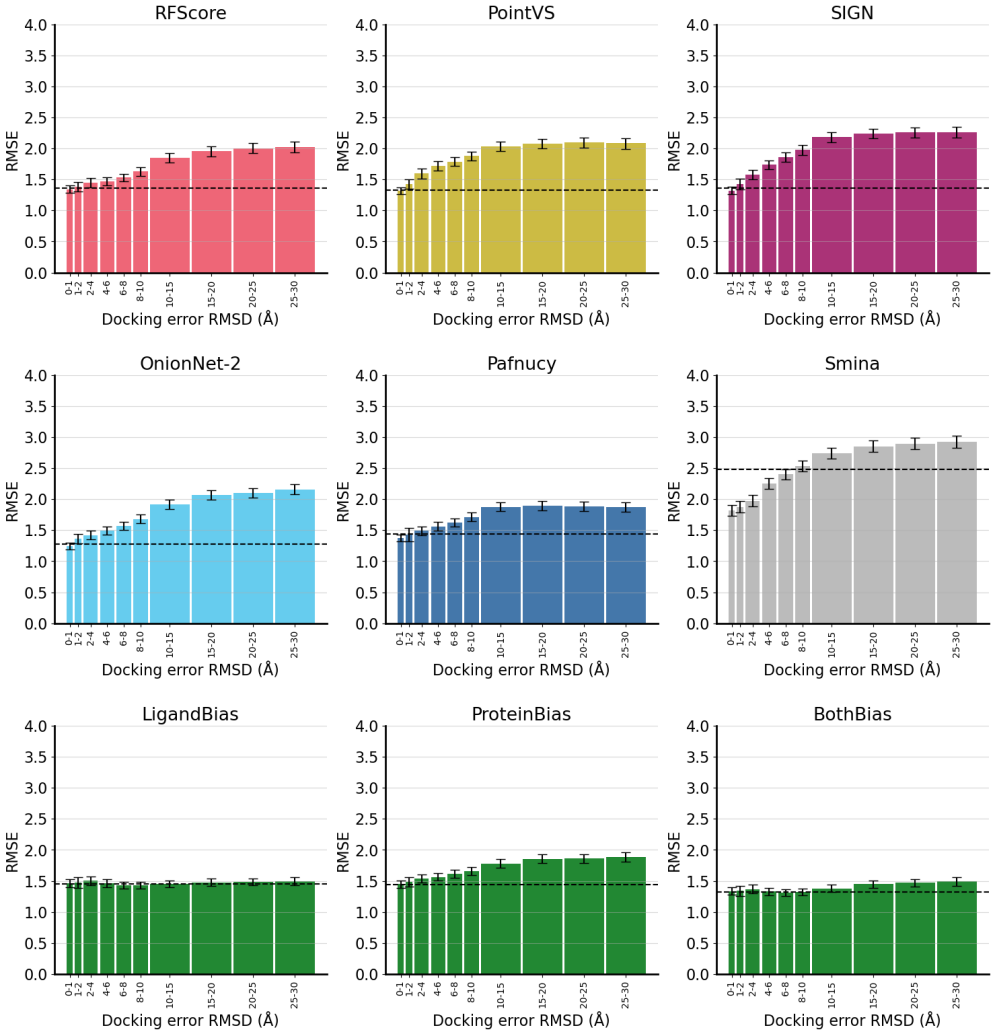


Figure A.10: RMSE between predicted and true pK values for protein-ligand complexes for the baseline models (LigandBias, ProteinBias and BothBias), a non-ML-based scoring function (Smina) and five commonly used MLBSFs (RFScore, PointVS, Pafnucy, SIGN and OnionNet-2), on different accuracy poses of CASF 2016 complexes. Accuracy on the crystal structures of CASF 2016 is shown as a dashed black line. Errors are the 95% confidence intervals from the bootstrapped RMSE (N=10000).

A.7 Accuracy of scoring functions and baseline models on differing docking accuracy versions of 2019 Holdout

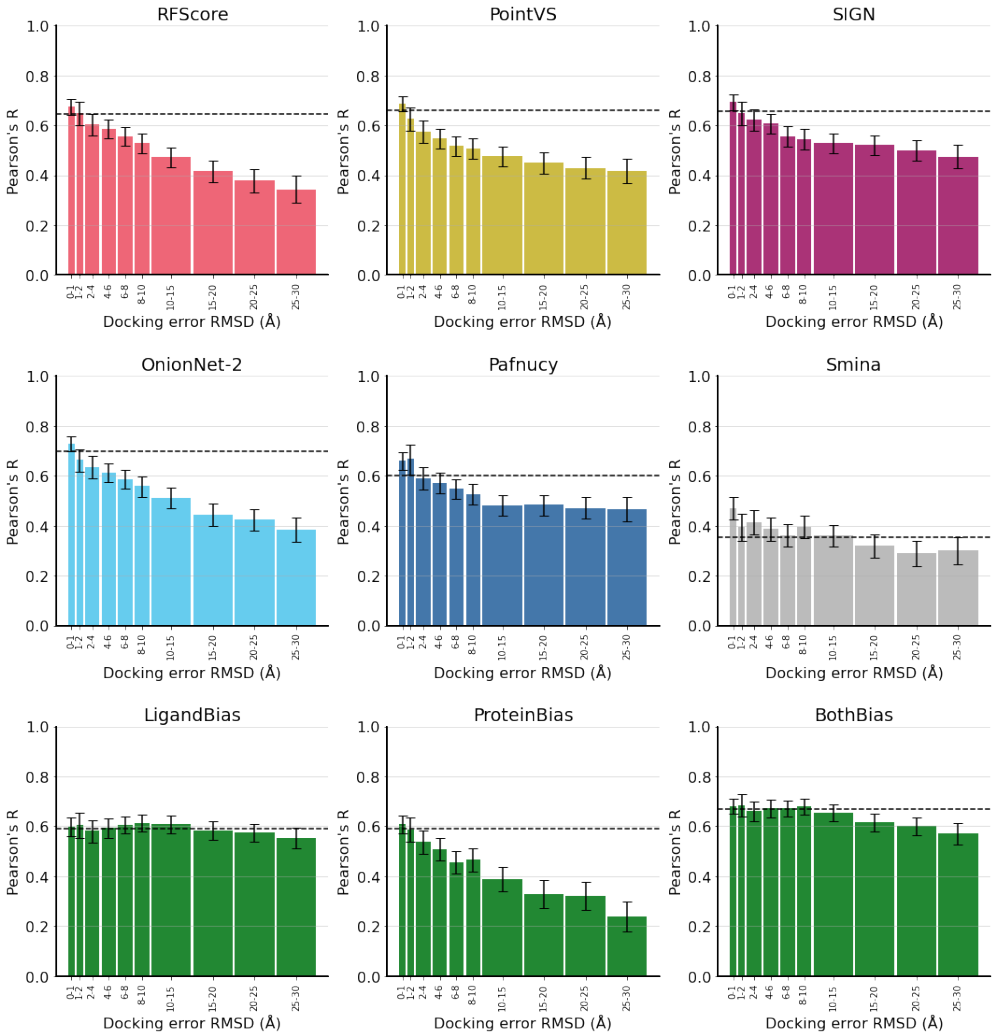


Figure A.11: Pearson's R between predicted and true pK values for protein-ligand complexes for the baseline models (LigandBias, ProteinBias and BothBias), a non-ML-based scoring function (Smina) and five commonly used MLBSFs (RFScore, PointVS, Pafnucy, SIGN and OnionNet-2), on different accuracy poses of 2019 Holdout complexes. Accuracy on the crystal structures of 2019 Holdout is shown as a dashed black line. Errors are the 95% confidence intervals from the bootstrapped Pearson's R (N=10000).

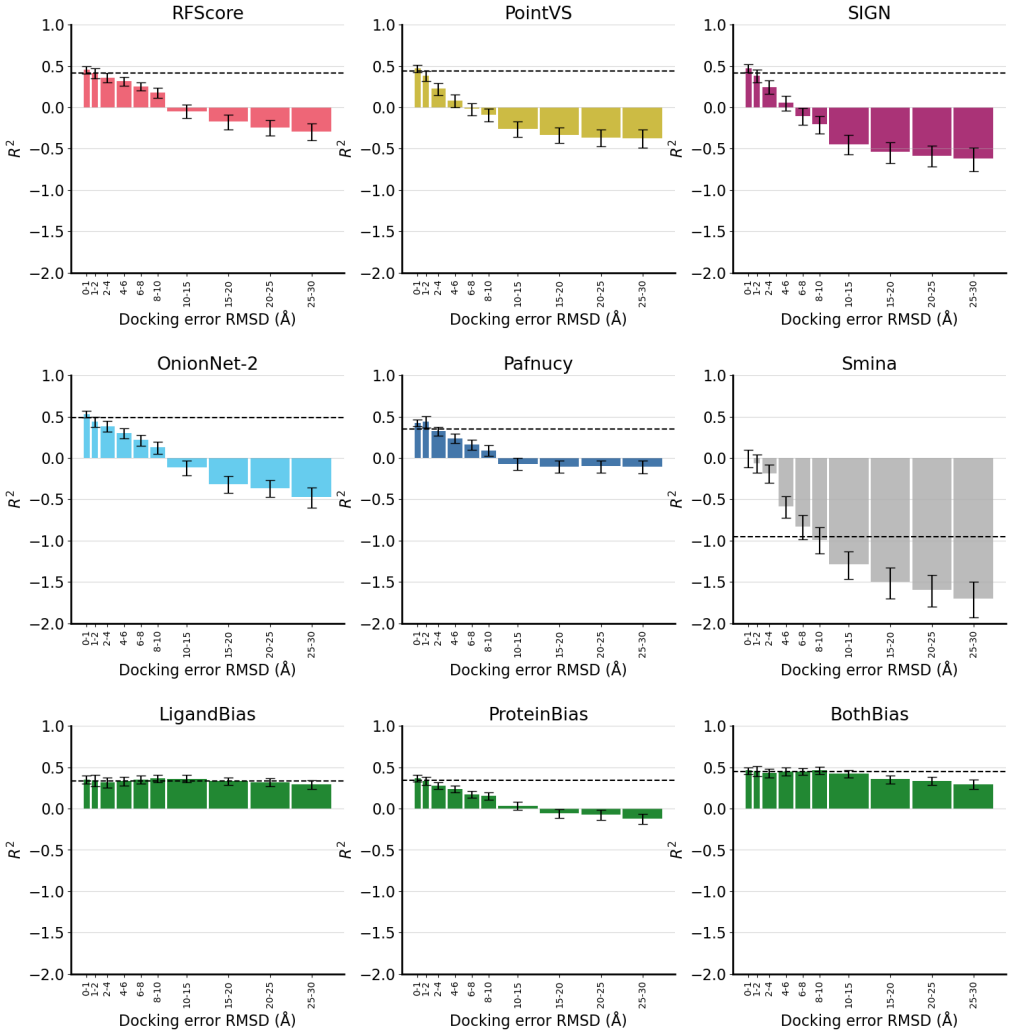


Figure A.12: R^2 between predicted and true pK values for protein-ligand complexes for the baseline models (LigandBias, ProteinBias and BothBias), a non-ML-based scoring function (Smina) and five commonly used MLBSFs (RFScore, PointVS, Pafnucy, SIGN and OnionNet-2), on different accuracy poses of 2019 Holdout complexes. Accuracy on the crystal structures of 2019 Holdout is shown as a dashed black line. Errors are the 95% confidence intervals from the bootstrapped R^2 (N=10000).

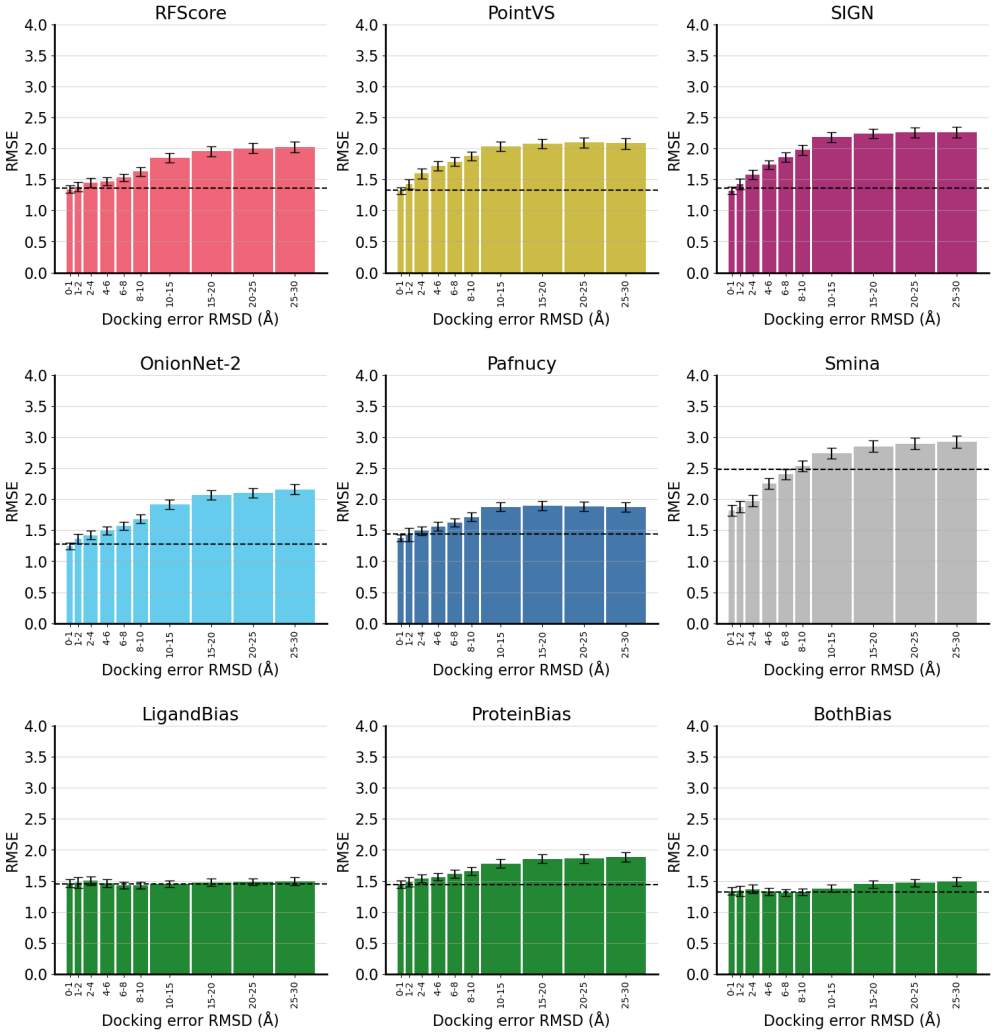


Figure A.13: RMSE between predicted and true pK values for protein-ligand complexes for the baseline models (LigandBias, ProteinBias and BothBias), a non-ML-based scoring function (Smina) and five commonly used MLBSFs (RFScore, PointVS, Pafnucy, SIGN and OnionNet-2), on different accuracy poses of 2019 Holdout complexes. Accuracy on the crystal structures of 2019 Holdout is shown as a dashed black line. Errors are the 95% confidence intervals from the bootstrapped RMSE (N=10000).

A.8 Accuracy of scoring functions and baseline models on differing docking accuracy versions of 0 Ligand Bias

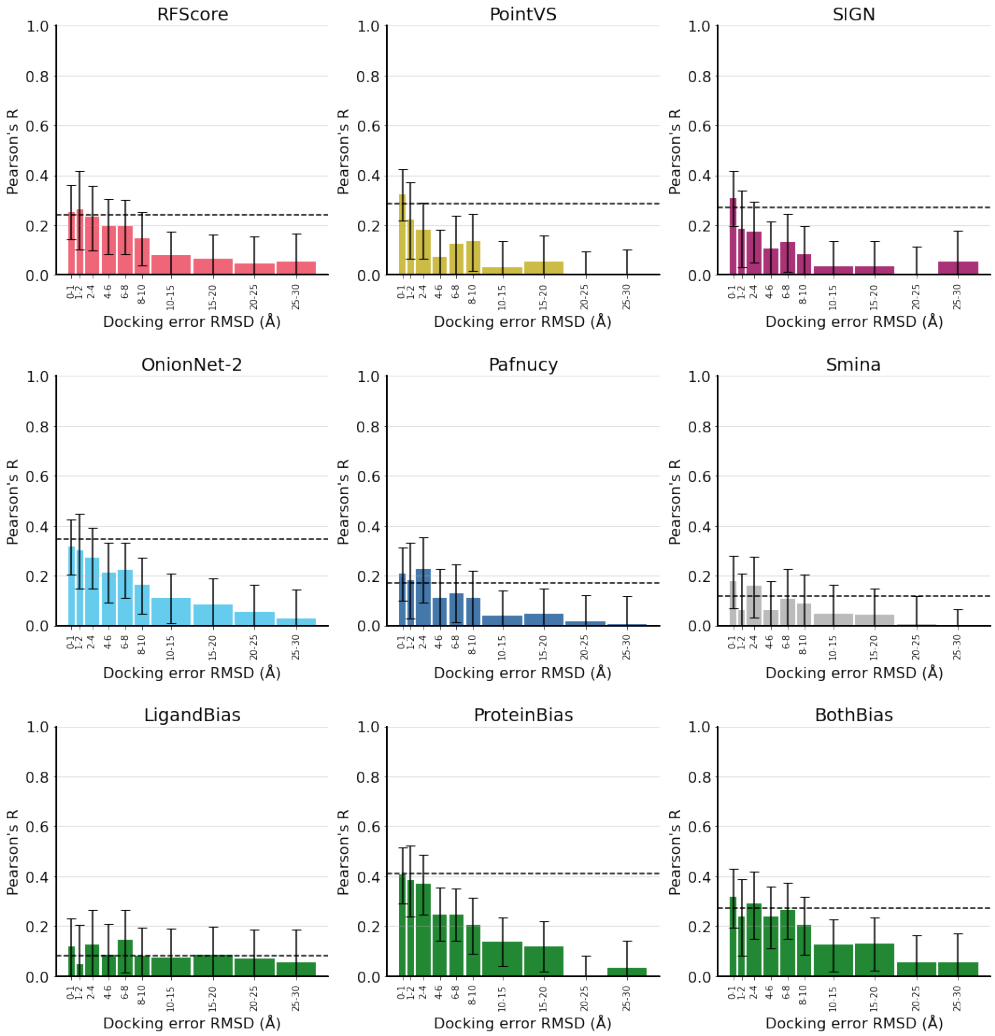


Figure A.14: Pearson's R between predicted and true pK values for protein-ligand complexes for the baseline models (LigandBias, ProteinBias and BothBias), a non-ML-based scoring function (Smina) and five commonly used MLBSFs (RFScore, PointVS, Pafnucy, SIGN and OnionNet-2), on different accuracy poses of 0 Ligand Bias complexes. Accuracy on the crystal structures of 0 Ligand Bias is shown as a dashed black line. Errors are the 95% confidence intervals from the bootstrapped Pearson's R (N=10000).

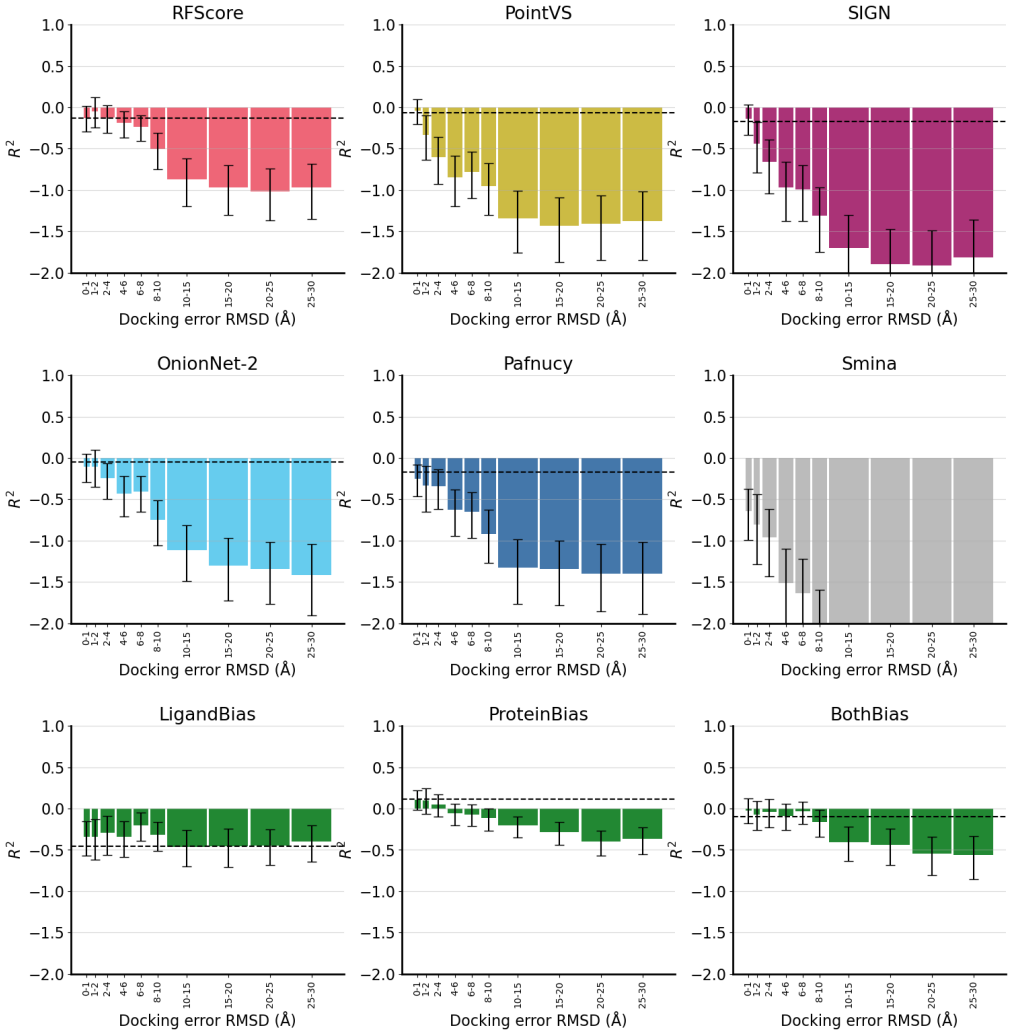


Figure A.15: R^2 between predicted and true pK values for protein-ligand complexes for the baseline models (LigandBias, ProteinBias and BothBias), a non-ML-based scoring function (Smina) and five commonly used MLBSFs (RFScore, PointVS, Pafnucy, SIGN and OnionNet-2), on different accuracy poses of 0 Ligand Bias complexes. Accuracy on the crystal structures of 0 Ligand Bias is shown as a dashed black line. Errors are the 95% confidence intervals from the bootstrapped R^2 (N=10000).

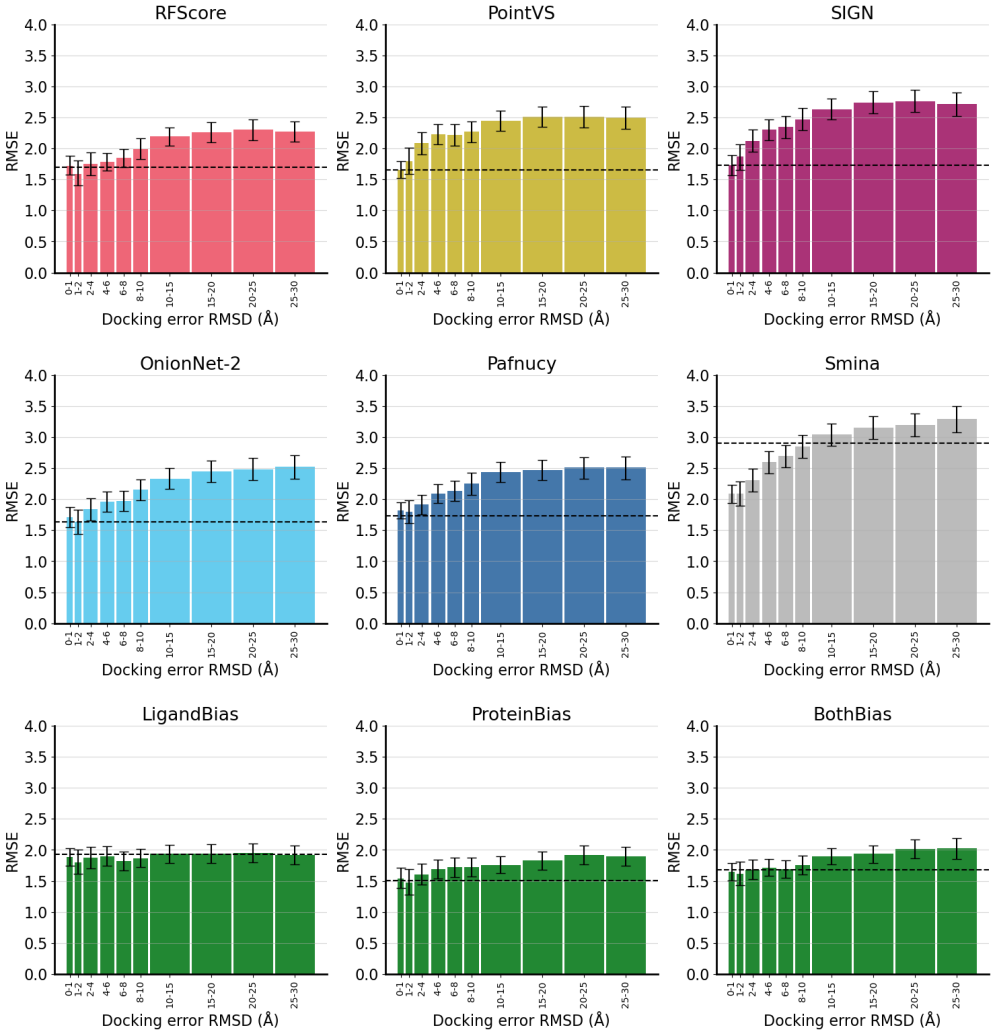


Figure A.16: RMSE between predicted and true pK values for protein-ligand complexes for the baseline models (LigandBias, ProteinBias and BothBias), a non-ML-based scoring function (Smina) and five commonly used MLBSFs (RFScore, PointVS, Pafnucy, SIGN and OnionNet-2), on different accuracy poses of 0 Ligand Bias complexes. Accuracy on the crystal structures of 0 Ligand Bias is shown as a dashed black line. Errors are the 95% confidence intervals from the bootstrapped RMSE (N=10000).

A.9 Accuracy of scoring functions and baseline models on progressively displaced ligands of CASF 2016

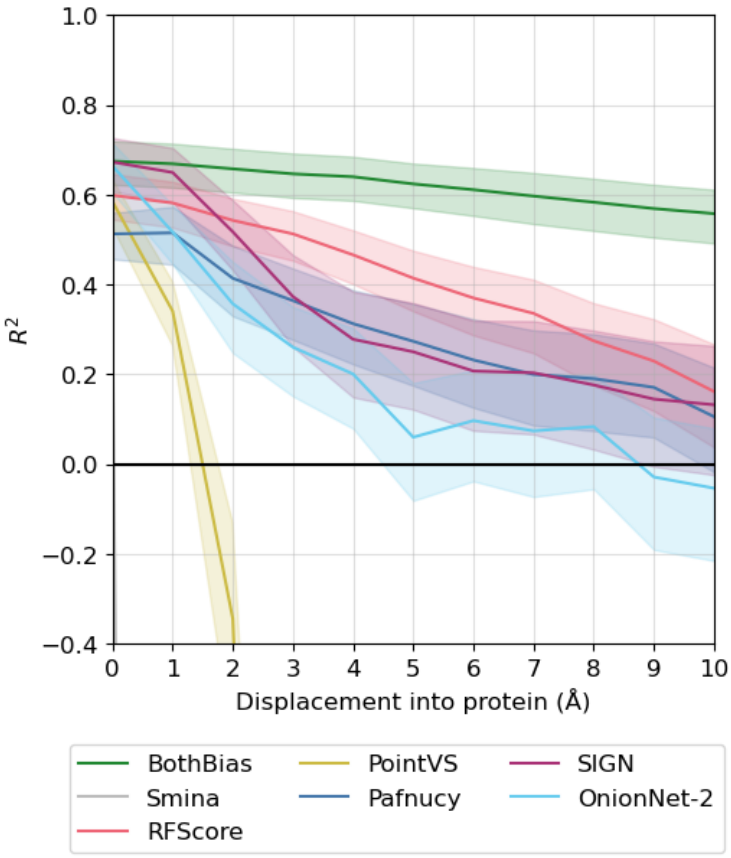


Figure A.17: R^2 between predicted and true pK values for protein-ligand com-plexes for the baseline model (BothBias), a non-ML-based scoring function (Smina) and five commonly used MLBSFs (RFScore, PointVS, Pafnucy, SIGN and OnionNet-2), on progressively displaced ligands into the protein originally from CASF 2016 crystal structures. Errors are the 95% confidence intervals from the bootstrapped R^2 (N=10000).

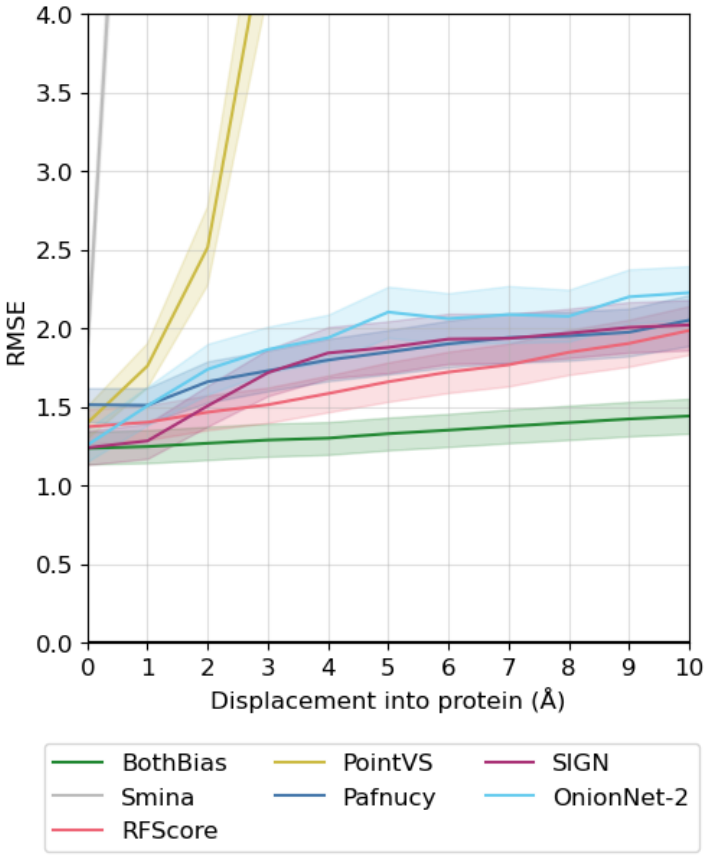


Figure A.18: RMSE between predicted and true pK values for protein-ligand complexes for the baseline model (BothBias), a non-ML-based scoring function (Smina) and five commonly used MLBSFs (RFScore, PointVS, Pafnucy, SIGN and OnionNet-2), on progressively displaced ligands into the protein originally from CASF 2016 crystal structures. Errors are the 95% confidence intervals from the bootstrapped RMSE (N=10000).

A.10 Accuracy of scoring functions and baseline models on progressively displaced ligands of 2019 Holdout

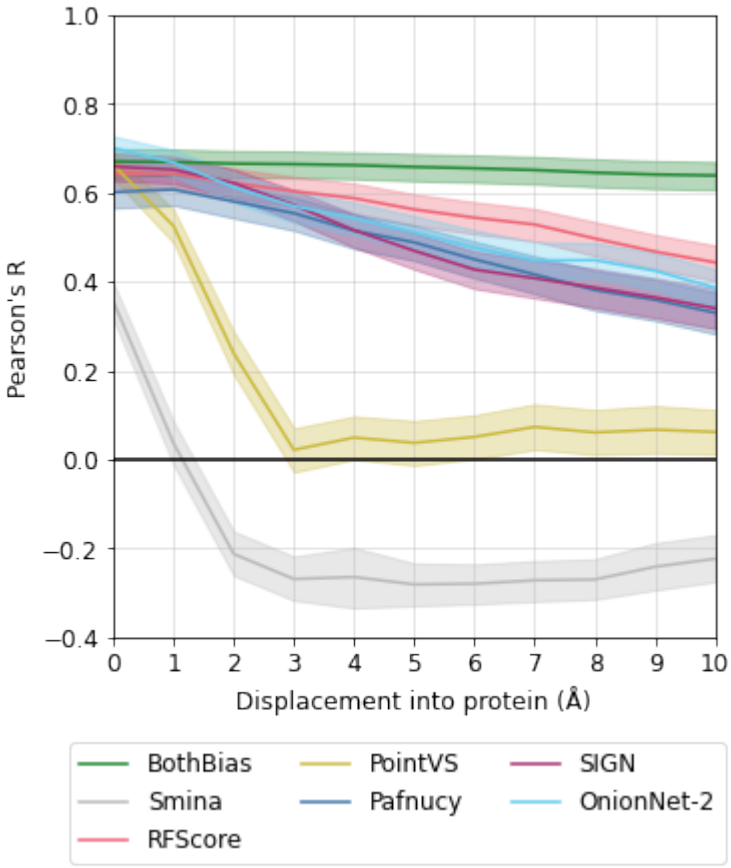


Figure A.19: Pearson's R between predicted and true pK values for protein-ligand complexes for the baseline model (BothBias), a non-ML-based scoring function (Smina) and five commonly used MLBSF (RFScore, PointVS, Pafnucy, SIGN and OnionNet-2), on progressively displaced ligands into the protein originally from 2019 Holdout crystal structures. Errors are the 95% confidence intervals from the bootstrapped Pearson's R (N=10000).

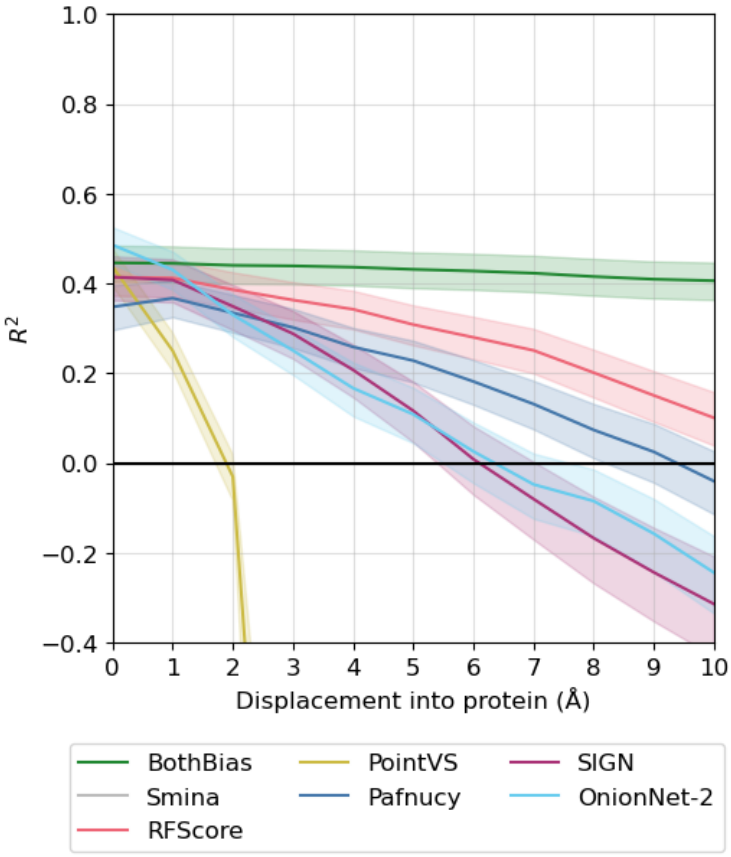


Figure A.20: R^2 between predicted and true pK values for protein-ligand com-plexes for the baseline model (BothBias), a non-ML-based scoring function (Smina) and five commonly used MLBSFs (RFScore, PointVS, Pafnucy, SIGN and OnionNet-2), on progressively displaced ligands into the protein originally from 2019 Holdout crystal structures. Errors are the 95% confidence intervals from the bootstrapped R^2 (N=10000).

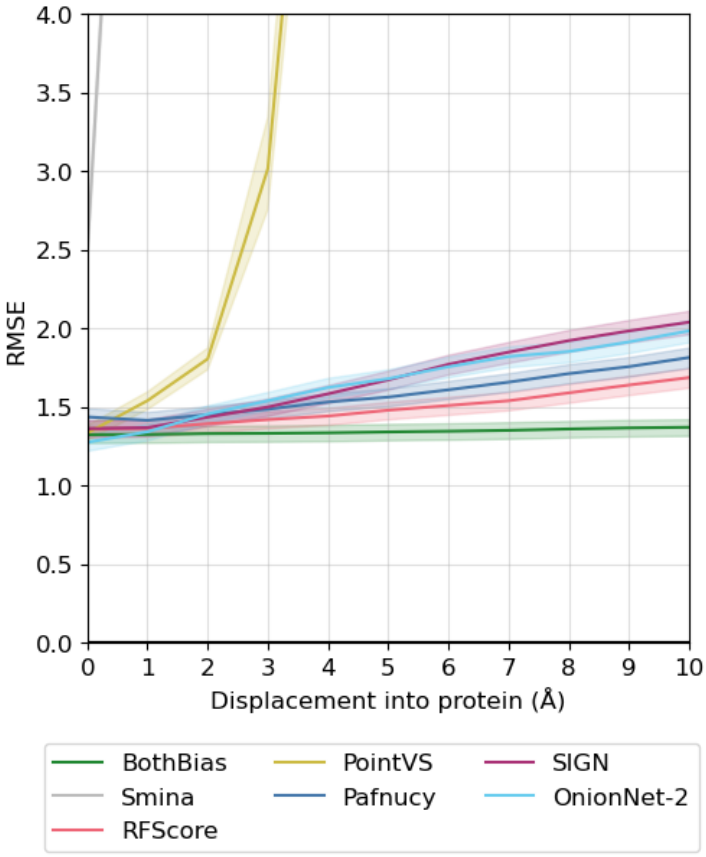


Figure A.21: RMSE between predicted and true pK values for protein-ligand complexes for the baseline model (BothBias), a non-ML-based scoring function (Smina) and five commonly used MLBSFs (RFScore, PointVS, Pafnucy, SIGN and OnionNet-2), on progressively displaced ligands into the protein originally from 2019 Holdout crystal structures. Errors are the 95% confidence intervals from the bootstrapped RMSE (N=10000).

A.11 Accuracy of scoring functions and baseline models on progressively displaced ligands of 0 Ligand Bias

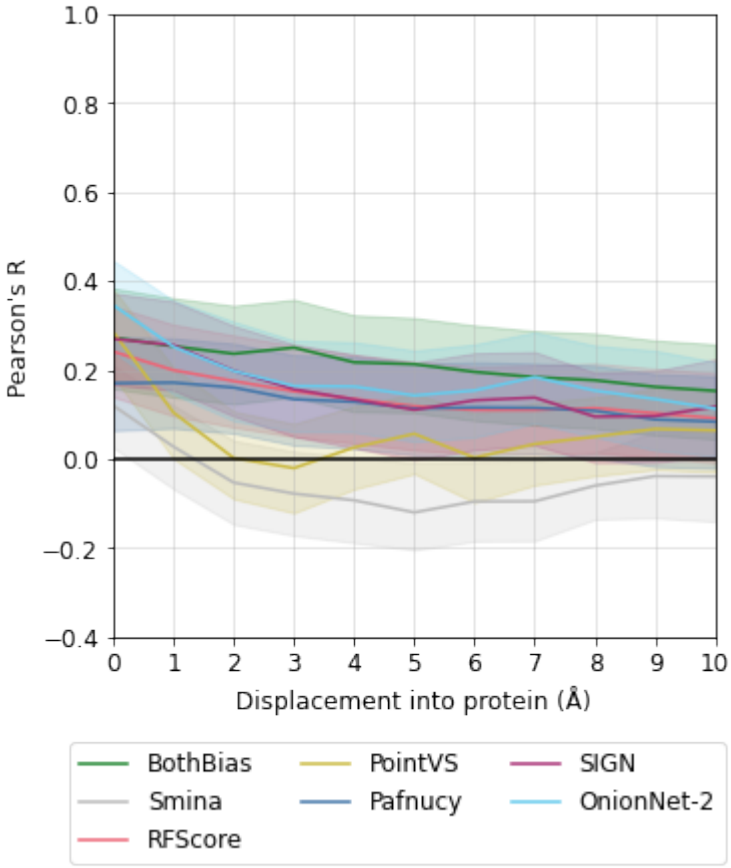


Figure A.22: Pearson's R between predicted and true pK values for protein-ligand com- plexes for the baseline model (BothBias), a non-ML-based scoring function (Smina) and five commonly used MLBSFs (RFScore, PointVS, Pafnucy, SIGN and OnionNet-2), on progressively displaced ligands into the protein originally from 0 Ligand Bias crystal structures. Errors are the 95% confidence intervals from the bootstrapped Pearson's R (N=10000).

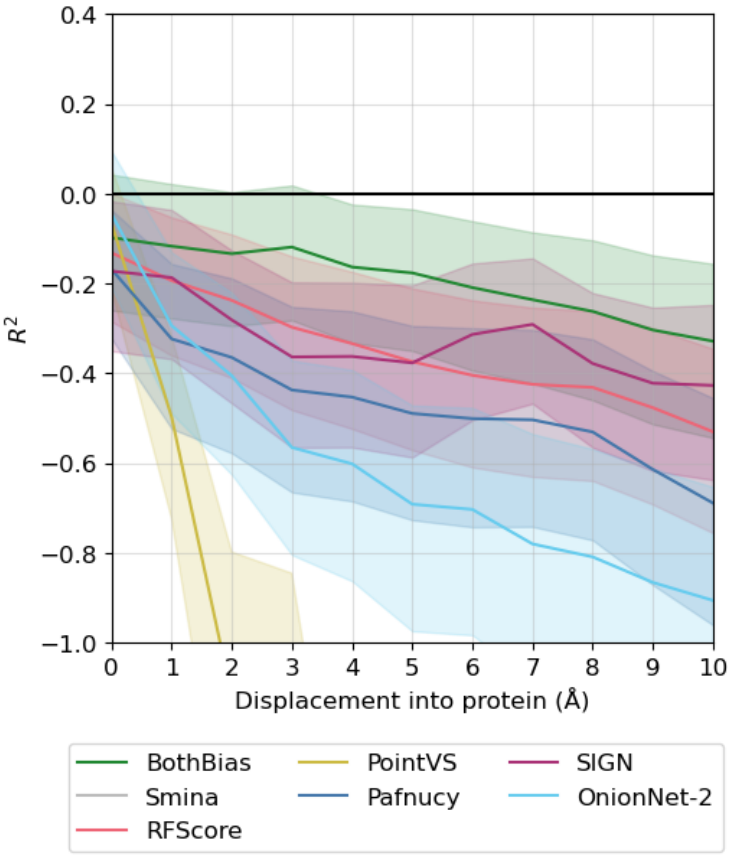


Figure A.23: R^2 between predicted and true pK values for protein-ligand com-plexes for the baseline models (LigandBias, ProteinBias and BothBias), a non-ML-based scoring function (Smina) and five commonly used MLBSFs (RFScore, PointVS, Pafnucy, SIGN and OnionNet-2), on progressively displaced ligands into the protein originally from 0 Ligand Bias crystal structures. Errors are the 95% confidence intervals from the bootstrapped R^2 (N=10000).

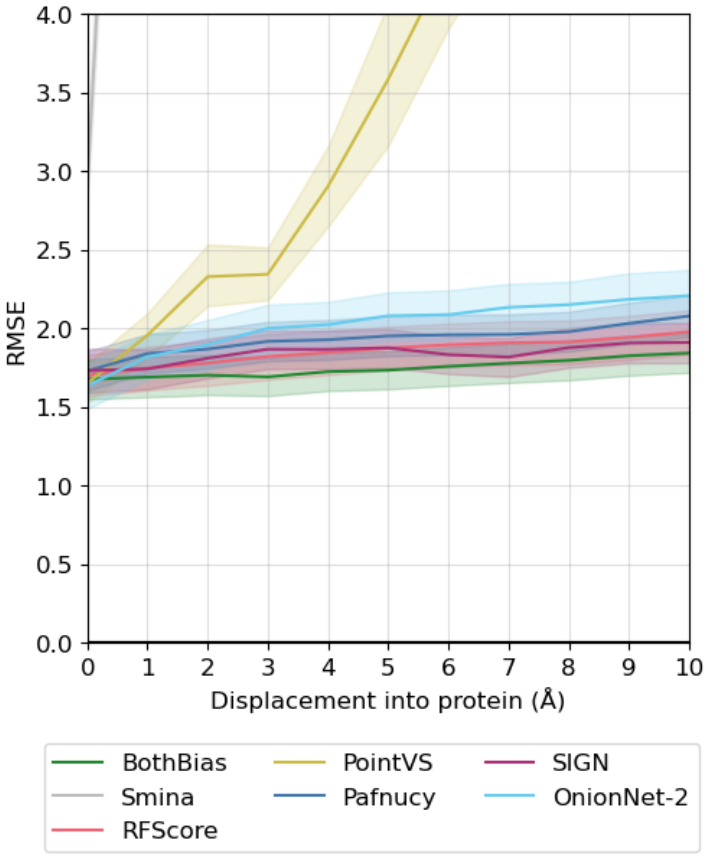


Figure A.24: RMSE between predicted and true pK values for protein-ligand complexes for the baseline models (LigandBias, ProteinBias and BothBias), a non-ML-based scoring function (Smina) and five commonly used MLBSFs (RFScore, PointVS, Pafnucy, SIGN and OnionNet-2), on progressively displaced ligands into the protein originally from 0 Ligand Bias crystal structures. Errors are the 95% confidence intervals from the bootstrapped RMSE (N=10000).

B

PoseTriager: improving pose classification robustness using data augmentation

B.1 Posebuster benchmark similarity subset PDB codes

B.1.1 0-30% (n=104)

6T88 6VTA 6XBO 6XM9 6YJA 6YRV 6YSP 6ZC3 6ZPB 7A9H 7AN5 7BNH 7C0U 7C3U 7C8Q 7CIJ 7CNQ 7CNS 7CTM 7CUO 7DKT 7E4L 7ED2 7EPV 7F8T 7FHA 7JMV 7KQU 7L00 7LJN 7LOE 7LZD 7M6K 7MFP 7MGT 7MOI 7MSR 7MWU 7N4W 7N7B 7N7H 7NFB 7O0N 7ODY 7OFK 7OMX 7OSO 7P2I 7PGX 7PJQ 7PK0 7PL1 7QE4 7QTA 7R3D 7R6J 7RC3 7RWS 7RZL 7SCW 7SDD 7TOM 7TUO 7U0U 7U3J 7UAW 7ULC 7UQ3 7UXS 7UY4 7V3N 7VKZ 7VYJ 7W05 7W06 7WCF 7WJB 7WKL 7WUX 7WUY 7X9K 7XJN 7XPO 7XQZ 7XRL 7YZU 7Z7F 7ZCC 7ZOC 7ZTL 8AIE 8AP0 8BTI 8C3N 8C5M 8CSD 8D19 8D5D 8DKO 8F4J 8F8E 8G6P 8GFD 8HFN

B.1.2 30-95% (n=89)

5SB2 6M73 6TW5 6TW7 6XCT 6XHT 6Z0R 6Z1C 6Z2C 6Z4N 6ZK5 7A1P 7AFX 7AKL 7B2C 7BCP 7CL8 7D5C 7ECR 7ELT 7ES1 7F51 7F5D 7FB7 7FRX 7JHQ 7KB1 7KM8 7KRU 7KZ9 7LMO 7M3H 7MMH 7MY1 7NF0 7NPL 7NSW 7NU0

7NUT 7OLI 7OP9 7OZ9 7OZC 7P1F 7P1M 7PRI 7PT3 7Q2B 7QF4 7R9N 7RKW 7ROR 7SIU 7SUC 7SZA 7T0D 7T3E 7TB0 7TBU 7TE8 7TH4 7TM6 7TXK 7UJ4 7UJ5 7UYB 7V43 7VB8 7VC5 7VQ9 7WQQ 7X5N 7XBV 7XG5 7Z1Q 7ZDY 7ZF0 7ZZW 8A1H 8AUH 8AY3 8B8H 8D39 8DP2 8EAB 8FO5 8G0V 8HO0 8SLG

B.1.3 95-100% (n=115)

5SIS 6WTN 6XG5 6Z14 7A9E 7B94 7BTT 7EBG 7FT9 7JXX 7K0V 7KC5 7LCU 7LOU 7M31 7MWN 7N03 7N6F 7NF3 7NLV 7NP6 7NR8 7P5T 7PIH 7POM 7QGP 7RSV 7UJF 7UMW 7WDT 7XFA 7ZHP 7ZU2 8A2D 8DHG 8EX2 8EXL 8FAV 5SAK 5SD5 6M2B 6YMS 6YQV 6YQW 6YR2 6YT6 6YYO 6ZAE 6ZCY 7BJJ 7BKA 7BMI 7CD9 7D6O 7DQL 7DUA 7JG0 7JY3 7L03 7L5F 7L7C 7LEV 7LT0 7MGY 7MYU 7N4N 7NGW 7NXO 7O1T 7OEO 7OFF 7OPG 7P4C 7PRM 7PUV 7Q25 7Q27 7Q5I 7QFM 7QHG 7QHL 7QPP 7R59 7R7R 7RH3 7RNI 7ROU 7SFO 7T1D 7THI 7TS6 7TSF 7TYP 7UAS 7USH 7UTW 7V14 7V3S 7VBU 7VWF 7WL4 7WPW 7WY1 7XI7 7Z2O 7ZL5 7ZXV 8AAU 8AEM 8AQL 8BOM 8CNH 8DSC 8EYE 8FLV

C

On the potential of ligand pocket design to synthetically expand the structural pocketome

C.1 Crystal benchmark PDB and CCD codes

C.1.1 Astex Diverse Set

1G9V-RQ3 1GKC-NFH 1GM8-SOX 1GPK-HUP 1HNN-SKF 1HP0-AD3 1HQ2-PH2 1HVY-D16 1HWI-115 1HWW-SWA 1IA1-TQ3 1IG3-VIB 1J3J-CP6 1JD0-AZM 1JJE-BYS 1JLA-TNK 1K3U-IAD 1KE5-LS1 1KZK-JE2 1L2S-STC 1L7F-BCZ 1LPZ-CMB 1LRH-NLA 1M2Z-DEX 1MEH-MOA 1MMV-3AR 1MZC-BNE 1N1M-A3M 1N2J-PAF 1N2V-BDI 1N46-PFA 1NAV-IH5 1OF1-SCT 1OF6-DTY 1OPK-P16 1OQ5-CEL 1OWE-675 1OYT-FSN 1P2Y-NCT 1P62-GEO 1PMN-984 1Q1G-MTI 1Q41-IXM 1Q4G-BFL 1R1H-BIR 1R55-097 1R58-AO5 1R9O-FLP 1S19-MC9 1S3V-TQD 1SG0-STL 1SJ0-E4D 1SQ5-PAU 1SQN-NDR 1T40-ID5 1T46-STI 1T9B-1CS 1TOW-CRZ 1TT1-KAI 1TZ8-DES 1U1C-BAU 1U4D-DBQ 1UML-FR4 1UNL-RRC 1UOU-CMU 1V0P-PVB 1V48-HA1 1V4S-MRK 1VCJ-IBA 1W1P-GIO 1W2G-THM 1X8X-TYR 1XM6-5RM 1XOQ-ROF 1XOZ-CIA 1Y6B-AAX 1YGC-905 1YQY-915 1YV3-BIT 1YVF-PH7 1YWR-LI9 1Z95-198 2BM2-PM2 2BR1-PFP 2BSM-BSM

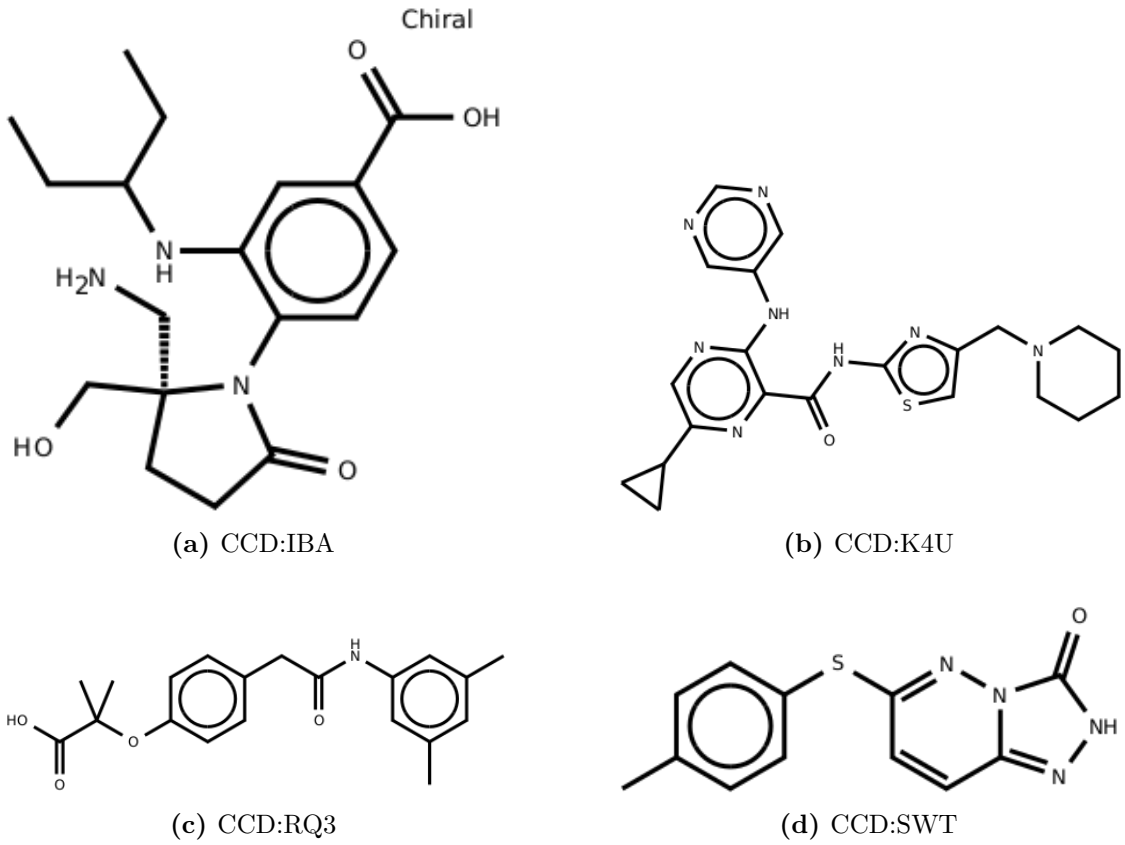
C.1.2 Runs N' Poses Cleaned Subset

7Q1O-9CI 7SES-97X 7WZV-7P8 7NYA-UUZ 5SIG-JP0 7S1D-648 7WC8-92S 7OMD-VK8 7WQM-4IV 5SHU-JIY 7UF9-NH0 7M0N-RLT 7X2Y-3HB 7EYU-0CC 5SH0-IZT 6ZG9-QK2 5SIZ-JWC 7SGK-9A9 5SGL-IXT 7XXN-7Y3 8E24-UA6 7F10-0BJ 7OME-VK8 8GUE-NRB 8DEG-SIQ 7AN8-RDH 8F4Q-XER 5SJ8-JYA 7XIS-E58 5SKT-KIY 7BC9-UR4 7VPE-7TU 7UFB-NJ0 5SHR-JIP 5SK5-JKO 7SER-97L 7B3H-SSW 8F1G-X8N 7XJ7-EGI 7QT2-7V7 5SGT-IYS 7EBA-J0U 7UX4-NWO 7RRE-6IV 7TXK-LW8 7NWC-UTQ 5SE0-IBJ 5SEE-IEN 7FEE-7IC 8E23-UAF 7S7X-QMR 7Z74-IKC 8B25-OSO 8GTM-0VI 7PFS-7OO 8ATY-O7O 7WQS-4ZI 7SLZ-9QU 8D3J-QEF 5SJL-K49 7XIF-TER 7XPV-GRJ 7VO5-81E 5SLG-NZJ 8DSM-TK0 7VI0-6YI 7WMF-9I9 8E5M-UKR 7VL7-7OU 7UKP-NJ9 7NUE-LSK 5SIH-JPF 5SI7-JML 5SI9-JN6 5SJ7-JY3 5SFL-IL9 7QTY-F0R 5SEN-IG9 8FDB-AGP 7Y06-IFQ 8AM2-N0C 8BI6-CIQ 8D3E-QDI 7VZY-83Z 5SKR-KII 7PAW-6IT 7QHE-C4I 8F1Z-X9B 8HE2-LMI 7Z57-IGB 8H7R-WZ0 8AU8-O8I 6ZFZ-QJT 5SJC-K0R 5SIY-JUC 7ERB-JBF 7Y03-IF4 5SGU-Z73 8HO5-XUN 7P84-EU9 7S1S-85K 7VQU-7UQ 7V34-5NF 7YPV-OI9 7B4X-SWT 7MXH-ZQ7 7ZJQ-JIK 5SF4-IJD 5SI8-JMU 5SKE-KF3 5SJ1-JWJ 7ZEG-IOO 7DIA-YMZ 7B3P-SUQ 7WZX-7PK 7PPQ-LSK 8F4S-XDU 8A9I-TSW 7EX3-IA3 8DSC-TIE 8B72-M25 7S84-8IL 7T1D-E7K 7W41-8B8 7FAK-1UI 7ZSO-JTC 7ZZS-KK0 7XM7-GAY 7Y02-IEZ 7VL0-PNG 7FD6-4QU 8DSH-QCT 7U1M-KYF 5SKO-KI4 7KBR-WAS 7ZGD-IUC 7N54-08V 7W2D-88E 8E1X-U9P 7W15-ERY 7BLW-AYR 7X9D-HRM 7PU8-4R7 7RWF-7TW 5SK6-KC9 5SE5-IEH 5SHC-JDU 5SEA-IF5 7SV5-D2Y 5SG5-IVP 7WDT-NGY 7XEF-D0I 8GTG-CW9 7E1N-ZHG 7E60-HX9 5SKL-KH6 7LPW-YBA 7NSN-MVL 8ETL-VND 7KBJ-WAV 8DSE-QCT 7TBD-I0L 8E62-MJZ 7F99-1UI 7XQZ-FPF 7QHD-C0I 5SDH-I81 8B8X-A8R 7X5Y-HZS 7ZOG-JE5 7RQ0-6I2 7ZG9-KO0 7T5R-F30 7X11-86I 7WLX-9GC 7R02-I62 7S1Q-84W 7YD2-IJC 7P1O-KDM 7RXO-80E 8D3H-QBC 8AI7-M8X 7ZST-JUF 5SF0-IIX 7MCJ-FD7 7XXC-I55 8AEV-LWU 7Q6W-93L 5S9Z-O3J 7S7T-QMR 7QZL-AML 7JNI-VFD 7DIJ-H8F 5SHK-JH6 7MWN-WI5 8B1V-ORZ 7B87-JFV 7UJE-I1F 7N9D-0XI 7DI7-CLQ 7X4T-2KH 8EIP-7ID 7RKE-5VP 8AQF-NUX 7VNH-BHF 7SQM-A5G 7L00-XCJ 7VHZ-6TI

7T5X-F4L 7WUY-76N 7XXM-I60 7DYQ-HR0 7UOH-NXF 7W1I-C8X 7UMO-NT6 7VL3-7OL 7AFZ-RBW 5S9Y-6SU 7N4Z-08N 7TE8-P0T 7X81-9LC 7L1P-R2A 5SI6-JME 7YOQ-IY0 7BMD-U3W 7ZGC-IUO 7UM4-NN6 7PWY-8EK 8EYM-16G 5SJW-K9O 5SH8-J9C 5SGP-IYD 7WMH-9II 7P8M-6IH 7OUM-1K8 5SK0-KAU 7PUQ-867 7YZU-DO7 8GTI-0JS 5SDY-IB4 7UFC-NJO 7S1R-85E 7Q1N-8IV 7YV4-JXY 5SKG-KFI 5SIQ-JRX 7N53-EV1 6ZG4-QK8 7Z5U-IFW 7VL6-7OQ 7F8U-1OE 5SFG-IKK 7UR3-OJ3 7SZC-DO0 7VZZ-83Z 7DMC-H9F 7ES1-JDF 7EI4-J3U 7SH0-GIY 8DSR-TWU 7WMI-9IC 8HFN-XGC 7YPU-I55 7B3X-SUT 7ZKW-IYY 5SI5-JM0 8DVO-U0I 7YXL-NDF 7PU9-85H 7B3G-SSQ 7QYN-I1X 7QEA-B9I 7T5Z-F6I 7ZZW-KKW 7VYP-82Z 5SJU-K4U 7EYT-0I3 7XUB-I6L 7VOC-7QO 7XXP-OI9 8GU1-1II 7DYW-HRL 7B3I-STK 7FAN-1TI 5SEX-II9 7ZZW-KZF 7XUA-I5X 8B43-OX8 7VQT-7UW 7ZGB-IUJ 5SKU-KJ3 5SEL-IG1 7F83-1KQ 7WMJ-9IL 8B8Z-Q33 8BHG-QMU 7PPY-7ZH 5SFR-IOI 7SV2-MWY

C.2 2D Ligands for Physically Invalid Depictions

C. On the potential of ligand pocket design to synthetically expand the structural pocketome



(a) CCD:IBA

(b) CCD:K4U

(c) CCD:RQ3

(d) CCD:SWT

Figure C.1: 2D structures of ligands for (a) FlowSite ligand implausibility, (b) LigandMPNN protein-ligand clash, (c) PocketGen side-chain packing implausibility, (d) LigandMPNN side-chain clashes.

References

Abanades, Brennan et al. (2023). 'ImmuneBuilder: Deep-Learning models for predicting the structures of immune proteins'. In: *Communications Biology* 6.1, p. 575.

Abramson, Josh et al. (2024). 'Accurate structure prediction of biomolecular interactions with AlphaFold 3'. In: *Nature*, pp. 1–3.

Achiam, Josh et al. (2023). 'Gpt-4 technical report'. In: *arXiv preprint arXiv:2303.08774*.

Ahdritz, Gustaf et al. (2024). 'OpenFold: Retraining AlphaFold2 yields new insights into its learning mechanisms and capacity for generalization'. In: *Nature Methods* 21.8, pp. 1514–1524.

Ajjarapu, Suchitra M et al. (2022). 'Ligand-based drug designing'. In: *Bioinformatics*. Elsevier, pp. 233–252.

Andersson, Shalini et al. (2009). 'Making medicinal chemistry more effective—application of Lean Sigma to improve processes, speed and quality'. In: *Drug Discovery Today* 14.11-12, pp. 598–604.

Anishchenko, Ivan et al. (2021). 'De novo protein design by deep network hallucination'. In: *Nature* 600.7889, pp. 547–552.

Arjovsky, Martin, Soumith Chintala and Léon Bottou (2017). 'Wasserstein generative adversarial networks'. In: *International Conference on Machine Learning*. PMLR, pp. 214–223.

Ashraf, S Neha et al. (2024). 'Hit me with your best shot: Integrated hit discovery for the next generation of drug targets'. In: *Drug Discovery Today* 29.10, p. 104143.

Austin, Christopher P (2021). 'Opportunities and challenges in translational science'. In: *Clinical and Translational Science* 14.5, pp. 1629–1647.

Azuaje, Gamar et al. (2023). 'Exploring the use of AI text-to-image generation to downregulate negative emotions in an expressive writing application'. In: *Royal Society Open Science* 10.1, p. 220238.

Babaoglu, Kerim et al. (2008). 'Comprehensive mechanistic analysis of hits from high-throughput and docking screens against β-lactamase'. In: *Journal of Medicinal Chemistry* 51.8, pp. 2502–2511.

Badrinarayan, Preethi and G Narahari Sastry (2011). 'Virtual high throughput screening in new lead identification'. In: *Combinatorial chemistry & high throughput screening* 14.10, pp. 840–860.

Baek, Minkyung et al. (2021). 'Accurate prediction of protein structures and interactions using a three-track neural network'. In: *Science* 373.6557, pp. 871–876.

Baell, Jonathan B and J Willem M Nissink (2018). 'Seven Year Itch: Pan-Assay Interference Compounds (PAINS) in 2017 Utility and Limitations'. In: *ACS Chemical Biology* 13.1, pp. 36–44.

Bahdanau, Dzmitry, Kyunghyun Cho and Yoshua Bengio (2014). 'Neural machine translation by jointly learning to align and translate'. In: *arXiv preprint arXiv:1409.0473*.

Ball, Philip (2020). 'Computer gleans chemical insight from lab notebook failures'. In: *Nature*.

Ballester, Pedro J and John BO Mitchell (2010). 'A machine learning approach to predicting protein–ligand binding affinity with applications to molecular docking'. In: *Bioinformatics* 26.9, pp. 1169–1175.

Ballester, Pedro J, Adrian Schreyer and Tom L Blundell (2014). 'Does a more precise chemical description of protein–ligand complexes lead to more accurate prediction of binding affinity?' In: *Journal of Chemical Information and Modeling* 54.3, pp. 944–955.

Ban, Thomas A (2006). 'The role of serendipity in drug discovery'. In: *Dialogues in Clinical Neuroscience* 8.3, pp. 335–344.

Bansal, Arpit et al. (2023). 'Universal guidance for diffusion models'. In: *IEEE/CVF*, pp. 843–852.

Batool, Maria, Bilal Ahmad and Sangdun Choi (2019). 'A structure-based drug discovery paradigm'. In: *International Journal of Molecular Sciences* 20.11, p. 2783.

Batzolis, Georgios et al. (2021). 'Conditional image generation with score-based diffusion models'. In: *arXiv preprint arXiv:2111.13606*.

Beltrán, Jesús et al. (2022). 'Rapid biosensor development using plant hormone receptors as reprogrammable scaffolds'. In: *Nature Biotechnology* 40.12, pp. 1855–1861.

Bennett, Nathaniel R et al. (2023). 'Improving de novo protein binder design with deep learning'. In: *Nature Communications* 14.1, p. 2625.

Bertoline, Letícia MF et al. (2023). 'Before and after AlphaFold2: An overview of protein structure prediction'. In: *Frontiers in Bioinformatics* 3, p. 1120370.

Betker, James et al. (2023). 'Improving image generation with better captions'. In: *Computer Science* 2, p. 3.

Bianco, Giulia et al. (2016). 'Covalent docking using autodock: Two-point attractor and flexible side chain methods'. In: *Protein Science* 25.1, pp. 295–301.

Bick, Matthew J et al. (2017). 'Computational design of environmental sensors for the potent opioid fentanyl'. In: *eLife* 6, e28909.

Bickerton, G Richard et al. (2012). 'Quantifying the chemical beauty of drugs'. In: *Nature Chemistry* 4.2, pp. 90–98.

Bissantz, Caterina, Gerd Folkers and Didier Rognan (2000). 'Protein-based virtual screening of chemical databases. 1. Evaluation of different docking/scoring combinations'. In: *Journal of Medicinal Chemistry* 43.25, pp. 4759–4767.

Blanco-Gonzalez, Alexandre et al. (2023). 'The role of AI in drug discovery: challenges, opportunities, and strategies'. In: *Pharmaceuticals* 16.6, p. 891.

Blaschke, Thomas et al. (2020). 'REINVENT 2.0: an AI tool for de novo drug design'. In: *Journal of Chemical Information and Modeling* 60.12, pp. 5918–5922.

Boettcher, Andreas et al. (2010). 'Fragment-based screening by biochemical assays: Systematic feasibility studies with trypsin and MMP12'. In: *Journal of Biomolecular Screening* 15.9, pp. 1029–1041.

Bordes, Antoine et al. (2013). 'Translating embeddings for modeling multi-relational data'. In: *Advances in Neural Information Processing Systems* 26.

Boutet, Emmanuel et al. (2007). 'UniProtKB/Swiss-Prot: the manually annotated section of the UniProt KnowledgeBase'. In: *Plant Bioinformatics*. Springer, pp. 89–112.

Boyles, Fergus, Charlotte M Deane and Garrett M Morris (2020). 'Learning from the ligand: using ligand-based features to improve binding affinity prediction'. In: *Bioinformatics* 36.3, pp. 758–764.

Boyles, Fergus, Charlotte M. Deane and Garrett M. Morris (2022). 'Learning from Docked Ligands: Ligand-Based Features Rescue Structure-Based Scoring Functions When Trained on Docked Poses'. In: *Journal of Chemical Information and Modeling* 62 (22), pp. 5329–5341.

Brandes, Nadav et al. (2022). 'ProteinBERT: a universal deep-learning model of protein sequence and function'. In: *Bioinformatics* 38.8, pp. 2102–2110.

Breiman, Leo (1996). 'Bagging predictors'. In: *Machine learning* 24.2, pp. 123–140.

— (2001). 'Random forests'. In: *Machine learning* 45.1, pp. 5–32.

Breiman, Leo et al. (2017). *Classification and regression trees*. Chapman and Hall/CRC.

Brenk, Ruth et al. (2008). 'Lessons learnt from assembling screening libraries for drug discovery for neglected diseases'. In: *ChemMedChem: Chemistry Enabling Drug Discovery* 3.3, pp. 435–444.

Brennan, Richard J et al. (2024). 'The state of the art in secondary pharmacology and its impact on the safety of new medicines'. In: *Nature Reviews Drug Discovery* 23.7, pp. 525–545.

Brian Houston, J and David J Carlile (1997). 'Prediction of hepatic clearance from microsomes, hepatocytes, and liver slices'. In: *Drug Metabolism Reviews* 29.4, pp. 891–922.

Brown, Scott P, Steven W Muchmore and Philip J Hajduk (2009). 'Healthy skepticism: assessing realistic model performance'. In: *Drug Discovery Today* 14.7-8, pp. 420–427.

Browne, Cameron B et al. (2012). 'A survey of monte carlo tree search methods'. In: *IEEE Transactions on Computational Intelligence and AI in Games* 4.1, pp. 1–43.

Bubeck, Sebastien et al. (2023). 'Sparks of artificial general intelligence: Early experiments with gpt-4'. In: *arXiv preprint arXiv:2303.12712*.

Burley, Stephen K et al. (2017). 'Protein Data Bank (PDB): the single global macromolecular structure archive'. In: *Protein Crystallography: Methods and Protocols*, pp. 627–641.

Buttenschoen, Martin, Garrett M Morris and Charlotte M Deane (2024). 'PoseBusters: AI-based docking methods fail to generate physically valid poses or generalise to novel sequences'. In: *Chemical Science* 15.9, pp. 3130–3139.

ByteDance et al. (2025). 'Protenix - Advancing Structure Prediction Through a Comprehensive AlphaFold3 Reproduction'. In: *BioRxiv*.

Cagiada, Matteo et al. (2021). 'Understanding the origins of loss of protein function by analyzing the effects of thousands of variants on activity and abundance'. In: *Molecular Biology and Evolution* 38.8, pp. 3235–3246.

Cao, Duanhua et al. (2025). 'SurfDock is a surface-informed diffusion generative model for reliable and accurate protein–ligand complex prediction'. In: *Nature Methods* 22.2, pp. 310–322.

Carlson, Heather A et al. (2016). 'CSAR 2014: a benchmark exercise using unpublished data from pharma'. In: *Journal of Chemical Information and Modeling* 56.6, pp. 1063–1077.

Carlson, Linda E, Amy Waller and Alex J Mitchell (2012). 'Screening for distress and unmet needs in patients with cancer: review and recommendations'. In: *Journal of Clinical Oncology* 30.11, pp. 1160–1177.

Carnero, Amancio (2006). 'High throughput screening in drug discovery'. In: *Clinical and Translational Oncology* 8.7, pp. 482–490.

Carter, Adrian J et al. (2019). 'Target 2035: probing the human proteome'. In: *Drug Discovery Today* 24.11, pp. 2111–2115.

Casewit, CJ, KS Colwell and AK Rappe (1992). 'Application of a universal force field to organic molecules'. In: *Journal of the American Chemical Society* 114.25, pp. 10035–10046.

Cerezo, Maria et al. (2025). 'The NHGRI-EBI GWAS Catalog: standards for reusability, sustainability and diversity'. In: *Nucleic Acids Research* 53.D1, pp. D998–D1005.

Cha, Y et al. (2018). 'Drug repurposing from the perspective of pharmaceutical companies'. In: *British Journal of Pharmacology* 175.2, pp. 168–180.

Chai-Discovery-Team et al. (2024). 'Chai-1: Decoding the molecular interactions of life'. In: *BioRxiv*, pp. 2024–10.

Chames, Patrick et al. (2009). 'Therapeutic antibodies: successes, limitations and hopes for the future'. In: *British Journal of Pharmacology* 157.2, pp. 220–233.

Chang, Max W et al. (2010). 'Virtual screening for HIV protease inhibitors: a comparison of AutoDock 4 and Vina'. In: *PloS One* 5.8, e11955.

Chaput, Ludovic et al. (2016). 'Benchmark of four popular virtual screening programs: construction of the active/decoy dataset remains a major determinant of measured performance'. In: *Journal of Cheminformatics* 8.1, pp. 1–17.

Chen, Kuang-Yui Michael, Daniel Keri and Patrick Barth (2020). 'Computational design of G Protein-Coupled Receptor allosteric signal transductions'. In: *Nature Chemical Biology* 16.1, pp. 77–86.

Chen, Lieyang et al. (2019). 'Hidden bias in the DUD-E dataset leads to misleading performance of deep learning in structure-based virtual screening'. In: *PloS One* 14.8, e0220113.

Chen, Ricky TQ and Yaron Lipman (2023). 'Flow matching on general geometries'. In: *arXiv preprint arXiv:2302.03660*.

Chen, Tianqi and Carlos Guestrin (2016). 'XGBoost: A scalable tree boosting system'. In: *Proceedings of the 22nd International Conference on Knowledge Discovery and Data Mining*, pp. 785–794.

Cheng, Tiejun et al. (2009). 'Comparative assessment of scoring functions on a diverse test set'. In: *Journal of Chemical Information and Modeling* 49.4, pp. 1079–1093.

Chithrananda, Seyone, Gabriel Grand and Bharath Ramsundar (2020). 'ChemBERTa: large-scale self-supervised pretraining for molecular property prediction'. In: *arXiv preprint arXiv:2010.09885*.

Cho, Yehlin et al. (2025). 'Boltzdesign1: Inverting all-atom structure prediction model for generalized biomolecular binder design'. In: *BioRxiv*, pp. 2025–04.

Cleves, Ann E and Ajay N Jain (2008). 'Effects of inductive bias on computational evaluations of ligand-based modeling and on drug discovery'. In: *Journal of Computer-aided Molecular Design* 22, pp. 147–159.

Cohen, Taco and Max Welling (2016). 'Group equivariant convolutional networks'. In: *International Conference on Machine Learning*. PMLR, pp. 2990–2999.

Cook, David et al. (2014). 'Lessons learned from the fate of AstraZeneca's drug pipeline: a five-dimensional framework'. In: *Nature Reviews Drug Discovery* 13.6, pp. 419–431.

Corso, Gabriele et al. (2022). 'Diffdock: Diffusion steps, twists, and turns for molecular docking'. In: *arXiv preprint arXiv:2210.01776*.

Corso, Gabriele et al. (2023). 'The Discovery of Binding Modes Requires Rethinking Docking Generalization'. In: *NeurIPS 2023 Generative AI and Biology (GenBio) Workshop*.

Cotet, Tudor-Stefan et al. (2025). 'Crowdsourced Protein Design: Lessons From the Adaptyv EGFR Binder Competition'. In: *BioRxiv*, pp. 2025–04.

Crawl, Common (2021). *Common Crawl Corpus*. Zenodo.

Croitoru, Florinel-Alin et al. (2023). 'Diffusion models in vision: A survey'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Daoud, Nour El-Huda et al. (2021). 'ADMET profiling in drug discovery and development: perspectives of in silico, in vitro and integrated approaches'. In: *Current Drug Metabolism* 22.7, pp. 503–522.

Dauparas, Justas et al. (2022). 'Robust deep learning–based protein sequence design using ProteinMPNN'. In: *Science* 378.6615, pp. 49–56.

Dauparas, Justas et al. (2023). 'Atomic context-conditioned protein sequence design using LigandMPNN'. In: *BioRxiv*, pp. 2023–12.

Delaney, John S (2004). 'ESOL: estimating aqueous solubility directly from molecular structure'. In: *Journal of Chemical Information and Computer Sciences* 44.3, pp. 1000–1005.

Dhariwal, Prafulla and Alexander Nichol (2021). 'Diffusion models beat gans on image synthesis'. In: *Advances in Neural Information Processing Systems* 34, pp. 8780–8794.

Diedrich, Konrad et al. (2023). 'PoseEdit: enhanced ligand binding mode communication by interactive 2D diagrams'. In: *Journal of Computer-Aided Molecular Design* 37.10, pp. 491–503.

Ding, Wenze, Kenta Nakai and Haipeng Gong (2022). 'Protein design via deep learning'. In: *Briefings in Bioinformatics* 23.3, bbac102.

Drews, Jurgen (2000). 'Drug discovery: a historical perspective'. In: *Science* 287.5460, pp. 1960–1964.

Dubey, Abhimanyu et al. (2024). 'The llama 3 herd of models'. In: *arXiv preprint arXiv:2407.21783*.

Dunn, Ian and David Ryan Koes (2024). 'Mixed continuous and categorical flow matching for 3d de novo molecule generation'. In: *arXiv preprint arXiv:2404*.

Durairaj, Janani et al. (2024). 'PLINDER: The protein-ligand interactions dataset and evaluation resource'. In: *BioRxiv*, pp. 2024–07.

Durrant, Jacob D and J Andrew McCammon (2011). 'NNScore 2.0: a neural-network receptor–ligand scoring function'. In: *Journal of Chemical Information and Modeling* 51.11, pp. 2897–2903.

Dutta, Shuchismita et al. (2014). 'Improving the representation of peptide-like inhibitor and antibiotic molecules in the Protein Data Bank'. In: *Biopolymers* 101.6, pp. 659–668.

Eberhardt, Jerome et al. (2021). 'AutoDock Vina 1.2. 0: new docking methods, expanded force field, and python bindings'. In: *Journal of Chemical Information and Modeling* 61.8, pp. 3891–3898.

Emmerich, Christoph H et al. (2021). 'Improving target assessment in biomedical research: the GOT-IT recommendations'. In: *Nature Reviews Drug Discovery* 20.1, pp. 64–81.

Errington, David et al. (2025). 'Assessing interaction recovery of predicted protein-ligand poses: D. Errington et al.' In: *Journal of Cheminformatics* 17.1, p. 76.

Estevam, Gabriella O et al. (2024). 'Mapping kinase domain resistance mechanisms for the MET receptor tyrosine kinase via deep mutational scanning'. In: *BioRxiv*.

Evans, Richard et al. (2021). 'Protein complex prediction with AlphaFold-Multimer'. In: *biorxiv*, pp. 2021–10.

Fearon, Daren et al. (2025). 'Accelerating Drug Discovery With High-Throughput Crystallographic Fragment Screening and Structural Enablement'. In: *Applied Research* 4.1, e202400192.

Ferla, Matteo (2021). *molecular_rectifier: A package to automatically correct RDKit-unsanitizable molecules*. GitHub repository.

Fernandez-Leiro, Rafael and Sjors HW Scheres (2016). 'Unravelling biological macromolecules with cryo-electron microscopy'. In: *Nature* 537.7620, pp. 339–346.

Ferreira, Leonardo G et al. (2015). 'Molecular docking and structure-based drug design strategies'. In: *Molecules* 20.7, pp. 13384–13421.

Finn, Robert D et al. (2014). 'Pfam: the protein families database'. In: *Nucleic Acids Research* 42.D1, pp. D222–D230.

Francoeur, Paul G et al. (2020). 'Three-dimensional convolutional neural networks and a cross-docked data set for structure-based drug design'. In: *Journal of Chemical Information and Modeling* 60.9, pp. 4200–4215.

Friesner, Richard A. et al. (2004). 'Glide: A New Approach for Rapid, Accurate Docking and Scoring. 1. Method and Assessment of Docking Accuracy'. In: *Journal of Medicinal Chemistry* 47 (7), pp. 1739–1749.

FTLOScience (2018). *The Process and Costs of Drug Development*. `https://ftloscience.com/process-costs-drug-development/`. Accessed: 2025-09-17.

Fukushima, Kunihiko (1980). 'Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position'. In: *Biological Cybernetics* 36.4, pp. 193–202.

Gainza, Pablo et al. (2020). 'Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning'. In: *Nature Methods* 17.2, pp. 184–192.

Gao, Wenhao and Connor W Coley (2020). 'The synthesizability of molecules proposed by generative models'. In: *Journal of Chemical Information and Modeling* 60.12, pp. 5714–5723.

Gao, Zhangyang et al. (2022). 'PiFold: Toward effective and efficient protein inverse folding'. In: *arXiv preprint arXiv:2209.12643*.

Gashaw, Isabella et al. (2011). 'What makes a good drug target?' In: *Drug Discovery Today* 16.23-24, pp. 1037–1043.

Gaulton, Anna et al. (2012). 'ChEMBL: a large-scale bioactivity database for drug discovery'. In: *Nucleic Acids Research* 40.D1, pp. D1100–D1107.

Geffner, Tomas et al. (2025). 'La-proteina: Atomistic protein generation via partially latent flow matching'. In: *arXiv preprint arXiv:2507.09466*.

Genheden, Samuel and ULF Ryde (2010). 'How to obtain statistically converged MM/GBSA results'. In: *Journal of Computational Chemistry* 31.4, pp. 837–846.

Gilmer, Justin et al. (2017). 'Neural message passing for quantum chemistry'. In: *International Conference on Machine Learning*. Pmlr, pp. 1263–1272.

Gilson, Michael K et al. (2016). 'BindingDB in 2015: a public database for medicinal chemistry, computational chemistry and systems pharmacology'. In: *Nucleic Acids Research* 44.D1, pp. D1045–D1053.

Giordanetto, Fabrizio et al. (2019). 'Fragment hits: what do they look like and how do they bind?' In: *Journal of Medicinal Chemistry* 62.7, pp. 3381–3394.

Gleeson, M Paul (2008). 'Generation of a set of simple, interpretable ADMET rules of thumb'. In: *Journal of Medicinal Chemistry* 51.4, pp. 817–834.

Goddard, Thomas D et al. (2018). 'UCSF ChimeraX: Meeting modern challenges in visualization and analysis'. In: *Protein Science* 27.1, pp. 14–25.

Gohlke, Holger, Manfred Hendlich and Gerhard Klebe (2000). 'Knowledge-based scoring function to predict protein-ligand interactions'. In: *Journal of Molecular Biology* 295.2, pp. 337–356.

Goodfellow, Ian et al. (2016). *Deep learning*. Vol. 1. 2. MIT press Cambridge.

Goodfellow, Ian J et al. (2014). 'Generative adversarial nets'. In: *Advances in Neural Information Processing Systems* 27.

Goodsell, David S and Arthur J Olson (1990). 'Automated docking of substrates to proteins by simulated annealing'. In: *Proteins: Structure, Function, and Bioinformatics* 8.3, pp. 195–202.

Goodsell, David S and Morris, Garrett M and Olson, Arthur J (1996). 'Automated docking of flexible ligands: applications of AutoDock'. In: *Journal of Molecular Recognition* 9.1, pp. 1–5.

Goverde, Casper A et al. (2023). 'De novo protein design by inversion of the AlphaFold structure prediction network'. In: *Protein Science* 32.6, e4653.

Gozalo-Brizuela, Roberto and Eduardo C Garrido-Merchán (2023). 'ChatGPT is not all you need. A State of the Art Review of large Generative AI models'. In: *GRACE: Global Review of AI Community Ethics* 1.1.

Guidi, Alessandra et al. (2015). 'Application of RNAi to genomic drug target validation in schistosomes'. In: *PLoS Neglected Tropical Diseases* 9.5, e0003801.

Guo, Qianrong, Saiveth Hernandez-Hernandez and Pedro J Ballester (2024). 'Scaffold splits overestimate virtual screening performance'. In: *International Conference on Artificial Neural Networks*. Springer, pp. 58–72.

Harris, Charles et al. (2023). 'Posecheck: Generative models for 3d structure-based drug design produce unrealistic poses'. In: *NeurIPS 2023 Generative AI and Biology (GenBio) Workshop*.

Hartshorn, Michael J et al. (2007). 'Diverse, high-quality test set for the validation of protein- ligand docking performance'. In: *Journal of Medicinal Chemistry* 50.4, pp. 726–741.

He, Kaiming et al. (2016). 'Deep residual learning for image recognition'. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778.

Hernández-Garrido, Carlos A and Norberto Sánchez-Cruz (2023). 'Experimental Uncertainty in Training Data for Protein-Ligand Binding Affinity Prediction Models'. In: *Artificial Intelligence in the Life Sciences* 4, p. 100087.

Hishigaki, Haretsugu and Satoru Kuhara (2011). 'hERGAPDbase: a database documenting hERG channel inhibitory potentials and APD-prolongation activities of chemical compounds'. In: *Database* 2011.

Ho, Jonathan, Ajay Jain and Pieter Abbeel (2020). 'Denoising diffusion probabilistic models'. In: *Advances in Neural Information Processing Systems* 33, pp. 6840–6851.

Ho, Jonathan and Tim Salimans (2022a). 'Classifier-free diffusion guidance'. In: *arXiv preprint arXiv:2207.12598*.

Ho, Jonathan et al. (2022b). 'Cascaded diffusion models for high fidelity image generation'. In: *Journal of Machine Learning Research* 23.47, pp. 1–33.

Hodge, Victoria and Jim Austin (2004). 'A survey of outlier detection methodologies'. In: *Artificial Intelligence Review* 22.2, pp. 85–126.

Holtzman, Ari et al. (2019). 'The curious case of neural text degeneration'. In: *arXiv preprint arXiv:1904.09751*.

Honda, Shion, Shoi Shi and Hiroki R Ueda (2019). 'Smiles transformer: Pre-trained molecular fingerprint for low data drug discovery'. In: *arXiv preprint arXiv:1911.04738*.

Hopf, Thomas A et al. (2014). 'Sequence co-evolution gives 3D contacts and structures of protein complexes'. In: *eLife* 3, e03430.

Hornak, Viktor et al. (2006). 'Comparison of multiple Amber force fields and development of improved protein backbone parameters'. In: *Proteins: Structure, Function, and Bioinformatics* 65.3, pp. 712–725.

Hornik, Kurt (1991). 'Approximation capabilities of multilayer feedforward networks'. In: *Neural networks* 4.2, pp. 251–257.

Hotelling, Harold (1933). 'Analysis of a complex of statistical variables into principal components.' In: *Journal of educational psychology* 24.6, p. 417.

Hsu, Chloe et al. (2022). 'Learning inverse folding from millions of predicted structures'. In: *International Conference on Machine Learning*. PMLR, pp. 8946–8970.

Hu, Liegi et al. (2005). 'Binding MOAD (mother of all databases)'. In: *Proteins: Structure, Function, and Bioinformatics* 60.3, pp. 333–340.

Hu, Xueping et al. (2022). 'Discovery of novel non-steroidal selective glucocorticoid receptor modulators by structure- and IGN-based virtual screening, structural optimization, and biological evaluation'. In: *European Journal of Medicinal Chemistry* 237, p. 114382.

Huang, Gao et al. (2017). 'Densely connected convolutional networks'. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708.

Huang, Lamei et al. (2021). 'KRAS mutation: from undruggable to druggable in cancer'. In: *Signal Transduction and Targeted Therapy* 6.1, p. 386.

Huang, Niu, Brian K. Shoichet and John J. Irwin (2006a). 'Benchmarking sets for molecular docking'. In: *Journal of Medicinal Chemistry* 49 (23), pp. 6789–6801.

Huang, Niu et al. (2006b). 'Molecular mechanics methods for predicting protein–ligand binding'. In: *Physical Chemistry Chemical Physics* 8.44, pp. 5166–5177.

Huang, Ruili et al. (2016). 'Tox21Challenge to build predictive models of nuclear receptor and stress response pathways as mediated by exposure to environmental chemicals and drugs'. In: *Frontiers in Environmental Science* 3, p. 85.

Huang, Sheng-You and Xiaoqin Zou (2006c). 'An iterative knowledge-based scoring function to predict protein–ligand interactions: II. Validation of the scoring function'. In: *Journal of Computational Chemistry* 27.15, pp. 1876–1882.

Hughes, James P et al. (2011). 'Principles of early drug discovery'. In: *British Journal of Pharmacology* 162.6, pp. 1239–1249.

Igashov, Ilia et al. (2024). 'Equivariant 3D-conditional diffusion model for molecular linker design'. In: *Nature Machine Intelligence*, pp. 1–11.

Ilari, Andrea and Carmelinda Savino (2008). 'Protein structure determination by x-ray crystallography'. In: *Bioinformatics: Data, Sequence Analysis and Evolution*, pp. 63–87.

Ildstad, Suzanne T and Charles H Evans Jr (2001). *Small clinical trials: issues and challenges*. National Academies Press.

Imrie, Fergus et al. (2020). 'Deep generative models for 3D linker design'. In: *Journal of Chemical Information and Modeling* 60.4, pp. 1983–1995.

Ingraham, John B et al. (2023). 'Illuminating protein space with a programmable generative model'. In: *Nature* 623.7989, pp. 1070–1078.

Jacobsson, Micael and Anders Karlén (2006). 'Ligand bias of scoring functions in structure-based virtual screening'. In: *Journal of Chemical Information and Modeling* 46.3, pp. 1334–1343.

Jadhav, Ajit et al. (2010). 'Quantitative analyses of aggregation, autofluorescence, and reactivity artifacts in a screen for inhibitors of a thiol protease'. In: *Journal of Medicinal Chemistry* 53.1, pp. 37–51.

Janela, Tiago and Jürgen Bajorath (2022). 'Simple nearest-neighbour analysis meets the accuracy of compound potency predictions using complex machine learning models'. In: *Nature Machine Intelligence* 4.12, pp. 1246–1255.

Jecklin, Matthias C et al. (2009). 'Label-free determination of protein–ligand binding constants using mass spectrometry and validation using surface plasmon resonance and isothermal titration calorimetry'. In: *Journal of Molecular Recognition* 22.4, pp. 319–329.

Jiang, Dejun et al. (2021a). 'Could graph neural networks learn better molecular representation for drug discovery? A comparison study of descriptor-based and graph-based models'. In: *Journal of Cheminformatics* 13 (1), pp. 1–23.

Jiang, Dejun et al. (2021b). 'InteractionGraphNet: a novel and efficient deep graph representation learning framework for accurate protein–ligand interaction predictions'. In: *Journal of Medicinal Chemistry* 64.24, pp. 18209–18232.

Jiang, Lin et al. (2008). 'De novo computational design of retro-aldol enzymes'. In: *Science* 319.5868, pp. 1387–1391.

Jiang, Yinjie et al. (2022). 'Artificial intelligence for retrosynthesis prediction'. In: *Engineering*.

Jiménez, José et al. (2018). 'K deep: protein–ligand absolute binding affinity prediction via 3d-convolutional neural networks'. In: *Journal of Chemical Information and Modeling* 58.2, pp. 287–296.

Jing, Bowen et al. (2022). 'Torsional Diffusion for Molecular Conformer Generation'. In: *Advances in Neural Information Processing Systems* 35.

Jones, Gareth et al. (1997). 'Development and validation of a genetic algorithm for flexible docking'. In: *Journal of Molecular Biology* 267.3, pp. 727–748.

Jorgensen, William L (2009). 'Efficient drug lead discovery and optimization'. In: *Accounts of Chemical Research* 42.6, pp. 724–733.

Jumper, John et al. (2021). 'Highly accurate protein structure prediction with AlphaFold'. In: *Nature* 596.7873, pp. 583–589.

Kairys, Visvaldas et al. (2019). 'Binding affinity in drug design: experimental and computational techniques'. In: *Expert Opinion on Drug Discovery* 14.8, pp. 755–768.

Kalliokoski, Tuomo et al. (2013). 'Comparability of mixed IC50 data–a statistical analysis'. In: *PloS One* 8.4, e61007.

Karlov, Dmitry S. et al. (2020). 'GraphDelta: MPNN Scoring Function for the Affinity Prediction of Protein-Ligand Complexes'. In: *ACS Omega* 5 (10), pp. 5150–5159.

Kausar, Samina and Andre O Falcao (2018). 'An automated framework for QSAR model building'. In: *Journal of Cheminformatics* 10.1, pp. 1–23.

Keserű, György M and Gergely M Makara (2006). 'Hit discovery and hit-to-lead approaches'. In: *Drug Discovery Today* 11.15-16, pp. 741–748.

Kim, Ryangguk and Jeffrey Skolnick (2008). 'Assessment of programs for ligand binding affinity prediction'. In: *Journal of Computational Chemistry* 29.8, pp. 1316–1331.

Kingma, Diederik P and Max Welling (2013). 'Auto-encoding variational bayes'. In: *arXiv preprint arXiv:1312.6114*.

Kipf, Thomas N and Max Welling (2016). 'Semi-supervised classification with graph convolutional networks'. In: *arXiv preprint arXiv:1609.02907*.

Klarner, Leo et al. (2022). 'Bias in the Benchmark: Systematic experimental errors in bioactivity databases confound multi-task and meta-learning algorithms'. In: *ICML 2022 2nd AI for Science Workshop*.

Klarner, Leo et al. (2023). 'Drug Discovery under Covariate Shift with Domain-Informed Prior Distributions over Functions'. In.

Klebe, Gerhard (2006). 'Virtual ligand screening: strategies, perspectives and limitations'. In: *Drug Discovery Today* 11.13-14, pp. 580–594.

Koes, David Ryan, Matthew P Baumgartner and Carlos J Camacho (2013). 'Lessons learned in empirical scoring with smina from the CSAR 2011 benchmarking exercise'. In: *Journal of Chemical Information and Modeling* 53.8, pp. 1893–1904.

Kola, Ismail and John Landis (2004). 'Can the pharmaceutical industry reduce attrition rates?' In: *Nature Reviews Drug Discovery* 3.8, pp. 711–716.

Kollman, Peter A et al. (2000). 'Calculating structures and free energies of complex molecules: combining molecular mechanics and continuum models'. In: *Accounts of chemical research* 33.12, pp. 889–897.

Korolev, Vadim et al. (2020). 'Graph Convolutional Neural Networks as "general-Purpose" Property Predictors: The Universality and Limits of Applicability'. In: *Journal of Chemical Information and Modeling* 60 (1), pp. 22–28.

Kramer, Christian and Peter Gedeck (2010). 'Leave-cluster-out cross-validation is appropriate for scoring functions derived from diverse protein data sets'. In: *Journal of Chemical Information and Modeling* 50.11, pp. 1961–1969.

Kramer, Christian et al. (2008). 'A composite model for hERG blockade'. In: *ChemMedChem: Chemistry Enabling Drug Discovery* 3.2, pp. 254–265.

Kramer, Christian et al. (2012). 'The experimental uncertainty of heterogeneous public K i data'. In: *Journal of Medicinal Chemistry* 55.11, pp. 5165–5173.

Krammer, André et al. (2005). 'LigScore: a novel scoring function for predicting binding affinities'. In: *Journal of Molecular Graphics and Modelling* 23.5, pp. 395–407.

Krishna, Rohith et al. (2024). 'Generalized biomolecular modeling and design with RoseTTAFold All-Atom'. In: *Science* 384.6693, eadl2528.

Krivák, Radoslav and David Hoksza (2018). 'P2Rank: machine learning based tool for rapid and accurate prediction of ligand binding sites from protein structure'. In: *Journal of Cheminformatics* 10.1, p. 39.

Krizhevsky, Alex, Ilya Sutskever and Geoffrey E Hinton (2012). 'Imagenet classification with deep convolutional neural networks'. In: *Advances in Neural Information Processing Systems* 25.

Kufareva, Irina, Andrey V Ilatovskiy and Ruben Abagyan (2012). 'Pocketome: an encyclopedia of small-molecule binding sites in 4D'. In: *Nucleic Acids Research* 40.D1, pp. D535–D540.

Kumar, Ashutosh, Arnout Voet and Kam YJ Zhang (2012). 'Fragment based drug design: from experimental to computational approaches'. In: *Current Medicinal Chemistry* 19.30, pp. 5128–5147.

Kuntz, Irwin D et al. (1982). 'A geometric approach to macromolecule-ligand interactions'. In: *Journal of Molecular Biology* 161.2, pp. 269–288.

Kuz'min, Victor E et al. (2011). 'Interpretation of QSAR models based on random forest methods'. In: *Molecular Informatics* 30.6-7, pp. 593–603.

Landrum, Greg (2023). *RDKit: Open-source cheminformatics*.

Landrum, Gregory A and Sereina Riniker (2024). 'Combining IC50 or K i Values from Different Sources Is a Source of Significant Noise'. In: *Journal of Chemical Information and Modeling* 64.5, pp. 1560–1567.

Leavy, Susan (2018). 'Gender bias in artificial intelligence: The need for diversity and gender theory in machine learning'. In: *1st International Workshop on Gender Equality in Software Engineering*, pp. 14–16.

LeCun, Yann et al. (2002). 'Gradient-based learning applied to document recognition'. In: *Proceedings of the IEEE* 86.11, pp. 2278–2324.

Lee, Nicol Turner (2018). 'Detecting racial bias in algorithms and machine learning'. In: *Journal of Information, Communication and Ethics in Society* 16.3, pp. 252–260.

Leman, Julia Koehler et al. (2020). 'Macromolecular modeling and design in Rosetta: recent methods and frameworks'. In: *Nature Methods* 17.7, pp. 665–680.

Leung, Susan et al. (2019). 'SuCOS is better than RMSD for evaluating fragment elaboration and docking poses'. In: *ChemRxiv*.

Li, Hongjian et al. (2015). 'Improving AutoDock Vina Using Random Forest: The Growing Accuracy of Binding Affinity Prediction by the Effective Exploitation of Larger Data Sets'. In: *Molecular Informatics* 34 (2-3), pp. 115–126.

Li, Ke and Craig M Crews (2022). 'PROTACs: past, present and future'. In: *Chemical Society Reviews* 51.12, pp. 5214–5236.

Li, Qimai, Zhichao Han and Xiao-Ming Wu (2018). 'Deeper insights into graph convolutional networks for semi-supervised learning'. In: *AAAI Conference on Artificial Intelligence*. Vol. 32. 1.

Li, Shuangli et al. (2021). 'Structure-aware interactive graph neural networks for the prediction of protein-ligand binding affinity'. In: *27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 975–985.

Li, Xuelian et al. (2024). 'A high-quality data set of protein–ligand binding interactions via comparative complex structure modeling'. In: *Journal of Chemical Information and Modeling* 64.7, pp. 2454–2466.

Li, Yan et al. (2014). 'Comparative assessment of scoring functions on an updated benchmark: 1. Compilation of the test set'. In: *Journal of Chemical Information and Modeling* 54.6, pp. 1700–1716.

Liao, Zhirui and You, Ronghui and Huang, Xiaodi and Yao, Xiaojun and Huang, Tao and Zhu, Shanfeng (2019). 'DeepDock: enhancing ligand-protein interaction prediction by a combination of ligand and structure information'. In: *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, pp. 311–317.

Lin, Zeming et al. (2023). 'Evolutionary-scale prediction of atomic-level protein structure with a language model'. In: *Science* 379.6637, pp. 1123–1130.

Linke, Pawel et al. (2016). 'An automated microscale thermophoresis screening approach for fragment-based lead discovery'. In: *Journal of Biomolecular Screening* 21.4, pp. 414–421.

Lipinski, Celio F et al. (2019). 'Advances and perspectives in applying deep learning for drug design and discovery'. In: *Frontiers in Robotics and AI* 6, p. 108.

Lipman, Yaron et al. (2022). 'Flow matching for generative modeling'. In: *arXiv preprint arXiv:2210.02747*.

Liu, Xiang et al. (2022). 'Dowker complex based machine learning (DCML) models for protein-ligand binding affinity prediction'. In: *PLoS Computational Biology* 18.4, e1009943.

Liu, Zhihai et al. (2014). 'PDB-wide collection of binding data: current status of the PDBbind database'. In: *Bioinformatics* 31.3, pp. 405–412.

Lloyd, Stuart (1982). 'Least squares quantization in PCM'. In: *IEEE Transactions on Information Theory* 28.2, pp. 129–137.

London, Nir, Dana Movshovitz-Attias and Ora Schueler-Furman (2010). 'The Structural Basis of Peptide-Protein Binding Strategies'. In: *Structure* 18.2, pp. 188–199.

Lowe, Daniel Mark (2012). 'Extraction of chemical structures and reactions from the literature'. PhD thesis. University of Cambridge.

Lu, Chong et al. (2022a). 'Systemic evolutionary chemical space exploration for drug discovery'. In: *Journal of Cheminformatics* 14.1, p. 19.

Lu, Wei et al. (2022b). 'Tankbind: Trigonometry-aware neural networks for drug-protein binding structure prediction'. In: *Advances in Neural Information Processing Systems* 35, pp. 7236–7249.

Ma, Junshui et al. (2015). 'Deep neural nets as a method for quantitative structure–activity relationships'. In: *Journal of Chemical Information and Modeling* 55.2, pp. 263–274.

Macarrón, Ricardo and Robert P Hertzberg (2011). 'Design and implementation of high throughput screening assays'. In: *Molecular Biotechnology* 47.3, pp. 270–285.

MacDermott-Opeskin, Hugo et al. (2025). 'A Computational Community Blind Challenge on Pan-Coronavirus Drug Discovery Data'. In: *ChemRxiv*.

Mahalmani, Vidya et al. (2022). *Translational research: Bridging the gap between preclinical and clinical research*.

Malisi, Christoph et al. (2012). 'Binding pocket optimization by computational protein design'. In: *PloS One* 7.12, e52505.

Maloney, Michael P. et al. (2023). 'Negative Data in Data Sets for Machine Learning Training'. In: *Organic Letters* 25 (17), pp. 2945–2947.

Mariani, Valerio et al. (2013). 'lDDT: a local superposition-free score for comparing protein structures and models using distance difference tests'. In: *Bioinformatics* 29.21, pp. 2722–2728.

Martins, Ines Filipa et al. (2012). 'A Bayesian approach to in silico blood-brain barrier penetration modeling'. In: *Journal of Chemical Information and Modeling* 52.6, pp. 1686–1697.

McCloskey, Kevin et al. (2020). 'Machine learning on DNA-encoded libraries: a new paradigm for hit finding'. In: *Journal of Medicinal Chemistry* 63.16, pp. 8857–8866.

McEwen, Leah and Fatima Mustafa (2023). 'WorldFAIR Chemistry: Making IUPAC Assets FAIR'. In: *Chemistry International* 45.1, pp. 14–17.

McNutt, Andrew T et al. (2021). 'GNINA 1.0: molecular docking with deep learning'. In: *Journal of Cheminformatics* 13.1, p. 43.

McNutt, Andrew T et al. (2025). 'GNINA 1.3: the next increment in molecular docking with deep learning'. In: *Journal of Cheminformatics* 17.1, p. 28.

Medicines and Healthcare products Regulatory Agency (2021). *Apply for a licence to market a medicine in the UK*. Accessed: 2025-09-16. URL: `https://www.gov.uk/guidance/apply-for-a-licence-to-market-a-medicine-in-the-uk`.

Meli, Rocco, Garrett M Morris and Philip C Biggin (2022). 'Scoring functions for protein-ligand binding affinity prediction using structure-based deep learning: a review'. In: *Frontiers in Bioinformatics* 2, p. 885983.

Meli, Rocco et al. (2021). 'Learning protein-ligand binding affinity with atomic environment vectors'. In: *Journal of Cheminformatics* 13 (1), pp. 1–19.

Menichetti, Roberto, Kiran H Kanekal and Tristan Bereau (2019). 'Drug–membrane permeability across chemical space'. In: *ACS Central Science* 5.2, pp. 290–298.

Merchant, Amil et al. (2023). 'Scaling deep learning for materials discovery'. In: *Nature*, pp. 1–6.

Metz, James T and Philip J Hajduk (2010). 'Rational approaches to targeted polypharmacology: creating and navigating protein–ligand interaction networks'. In: *Current Opinion in Chemical Biology* 14.4, pp. 498–504.

Meyers, Joshua, Benedek Fabian and Nathan Brown (2021). 'De novo molecular design and generative models'. In: *Drug Discovery Today* 26.11, pp. 2707–2715.

Miao, Zhichao, Yang Cao and Taijiao Jiang (2011). 'RASP: rapid modeling of protein side chain conformations'. In: *Bioinformatics* 27.22, pp. 3117–3122.

Midlam, Cody (2020). 'Status of Biologic Drugs in Modern Therapeutics-Targeted Therapies vs. Small Molecule Drugs'. In: *Biologics, Biosimilars, and Biobetters: An Introduction for Pharmacists, Physicians, and Other Health Practitioners*, pp. 31–46.

Mitchell, Alex L et al. (2020). 'MGnify: the microbiome analysis resource in 2020'. In: *Nucleic Acids Research* 48.D1, pp. D570–D578.

Mitchell, Tom M (1997). 'Does machine learning really work?' In: *AI Magazine* 18.3, pp. 11–11.

Mitchell, John BO (2014). 'Machine learning methods in chemoinformatics'. In: *Wiley Interdisciplinary Reviews: Computational Molecular Science* 4.5, pp. 468–481.

Mlinarić, Ana, Martina Horvat and Vesna Šupak Smolčić (2017). 'Dealing with the positive publication bias: Why you should really publish your negative results'. In: *Biochemia Medica* 27.3, pp. 447–452.

Mnih, Volodymyr et al. (2015). 'Human-level control through deep reinforcement learning'. In: *Nature* 518.7540, pp. 529–533.

Mobley, David L. and Michael K. Gilson (2017). 'Predicting Binding Free Energies: Frontiers and Benchmarks'. In: *Annual review of biophysics* 46, pp. 531–558.

Montgomery, Douglas C, Elizabeth A Peck and G Geoffrey Vining (2021). *Introduction to linear regression analysis*. John Wiley & Sons.

Moon, Seokhyun et al. (2022). 'PIGNet: a physics-informed deep learning model toward generalized drug–target interaction predictions'. In: *Chemical Science* 13.13, pp. 3661–3673.

Morehead, Alex and Jianlin Cheng (2025). 'Flowdock: Geometric flow matching for generative protein-ligand docking and affinity prediction'. In: *ArXiv*, arXiv–2412.

Morphy, Richard (2010). 'Selectively nonselective kinase inhibition: striking the right balance'. In: *Journal of Medicinal Chemistry* 53.4, pp. 1413–1437.

Morris, Garrett M and Marguerita Lim-Wilby (2008). 'Molecular docking'. In: *Molecular Modeling of Proteins*. Springer, pp. 365–382.

Morris, Garrett M et al. (2009). 'AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility'. In: *Journal of Computational Chemistry* 30.16, pp. 2785–2791.

Muegge, Ingo and Yuan Hu (2023). 'Recent advances in alchemical binding free energy calculations for drug discovery'. In: *ACS Medicinal Chemistry Letters* 14.3, pp. 244–250.

Muegge, Ingo and Yvonne C Martin (1999). 'A general and fast scoring function for protein- ligand interactions: a simplified potential approach'. In: *Journal of Medicinal Chemistry* 42.5, pp. 791–804.

Murray, Christopher W, Marcel L Verdonk and David C Rees (2012). 'Experiences in fragment-based drug discovery'. In: *Trends in Pharmacological Sciences* 33.5, pp. 224–232.

Mysinger, Michael M et al. (2012). 'Directory of useful decoys, enhanced (DUD-E): better ligands and decoys for better benchmarking'. In: *Journal of Medicinal Chemistry* 55.14, pp. 6582–6594.

Nagata, Ken, Arlo Randall and Pierre Baldi (2012). 'SIDEpro: A novel machine learning approach for the fast and accurate prediction of side-chain conformations'. In: *Proteins: Structure, Function, and Bioinformatics* 80.1, pp. 142–153.

Nidhi, Sweta et al. (2021). 'Novel CRISPR–Cas systems: an updated review of the current achievements, applications, and future research perspectives'. In: *International Journal of Molecular Sciences* 22.7, p. 3327.

Noske, Jakob et al. (2023). 'PocketOptimizer 2.0: A modular framework for computer-aided ligand-binding design'. In: *Protein Science* 32.1, e4516.

Notin, Pascal et al. (2023). 'Proteingym: Large-scale benchmarks for protein fitness prediction and design'. In: *Advances in Neural Information Processing Systems* 36, pp. 64331–64379.

Notin, Pascal et al. (2024). 'Machine learning for functional protein design'. In: *Nature Biotechnology* 42.2, pp. 216–228.

O'Boyle, Noel M et al. (2011). 'Open Babel: An open chemical toolbox'. In: *Journal of Cheminformatics* 3.1, pp. 1–14.

Olsen, Tobias H, Fergus Boyles and Charlotte M Deane (2022a). 'Observed Antibody Space: A diverse database of cleaned, annotated, and translated unpaired and paired antibody sequences'. In: *Protein Science* 31.1, pp. 141–146.

Olsen, Tobias H, Iain H Moal and Charlotte M Deane (2022b). 'AbLang: an antibody language model for completing antibody sequences'. In: *Bioinformatics Advances* 2.1, vbac046.

Onakpoya, Igho J (2018). 'Rare adverse events in clinical trials: understanding the rule of three'. In: *BMJ Evidence-Based Medicine* 23.1, pp. 6–6.

OpenBind (2025). *OpenBind: Generating foundational structural biology data to power the next era of AI/ML for drug discovery*. `https://openbind.uk/`. Accessed: 2025-09-19.

Pacesa, Martin et al. (2024). 'BindCraft: one-shot design of functional protein binders'. In: *BioRxiv*, pp. 2024–09.

Park, Guiyoung et al. (2024). 'Replacing animal testing with stem cell-organoids: advantages and limitations'. In: *Stem Cell Reviews and Reports* 20.6, pp. 1375–1386.

Park, Jung Woo et al. (2021). 'mRNA vaccines for COVID-19: what, why and how'. In: *International Journal of Biological Sciences* 17.6, p. 1446.

Passaro, Saro et al. (2025). 'Boltz-2: Towards accurate and efficient binding affinity prediction'. In: *BioRxiv*, pp. 2025–06.

Passini, Elisa et al. (2017). 'Human in silico drug trials demonstrate higher accuracy than animal models in predicting clinical pro-arrhythmic cardiotoxicity'. In: *Frontiers in Physiology* 8, p. 668.

Paul, Steven M et al. (2010). 'How to improve R&D productivity: the pharmaceutical industry's grand challenge'. In: *Nature Reviews Drug Discovery* 9.3, pp. 203–214.

PDB (2023). *RCSB PDB: Programmatic Access – File Download Services – Sequence Clusters Data*. `https://www.rcsb.org/docs/programmatic-access/file-download-services`. Accessed: 2023-06-28.

Pedregosa, Fabian et al. (2011). 'Scikit-learn: Machine learning in Python'. In: *Journal of Machine Learning Research* 12.Oct, pp. 2825–2830.

Pinheiro, Pedro de Sena Murteira, Lucas Silva Franco and Carlos Alberto Manssour Fraga (2023). 'The magic methyl and its tricks in drug discovery and development'. In: *Pharmaceuticals* 16.8, p. 1157.

Plainer, Michael et al. (2023). 'DiffDock-Pocket: Diffusion for Pocket-Level Docking with Sidechain Flexibility'. In: *NeurIPS 2023 Workshop on New Frontiers of AI for Drug Discovery and Development*.

Pletnev, Igor et al. (2012). 'InChIKey collision resistance: an experimental testing'. In: *Journal of Cheminformatics* 4, pp. 1–9.

Porter, Dale (2023). 'Why Failing Faster Is the Key to Accelerating Drug Discovery.' In: *Evidence-Based Oncology* 29.8, NA–NA.

Prior, Marguerite et al. (2014). 'Back to the future with phenotypic screening'. In: *ACS Chemical Neuroscience* 5.7, pp. 503–513.

Qi, Charles R et al. (2017). 'Pointnet: Deep learning on point sets for 3d classification and segmentation'. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 652–660.

Radford, Alec, Luke Metz and Soumith Chintala (2015). 'Unsupervised representation learning with deep convolutional generative adversarial networks'. In: *arXiv preprint arXiv:1511.06434*.

Radford, Alec et al. (2018). 'Improving language understanding by generative pre-training'. In: *ArXiv*.

Ragoza, Matthew et al. (2017). 'Protein–ligand scoring with convolutional neural networks'. In: *Journal of Chemical Information and Modeling* 57.4, pp. 942–957.

Ralston, Sea (2017). 'Pre-development attrition of pharmaceuticals: how to identify the bad actors early'. In: *Toxicol. Sci* 150, p. 2323.

Ramesh, Aditya et al. (2021). 'Zero-shot text-to-image generation'. In: *International Conference on Machine Learning*. PMLR, pp. 8821–8831.

Ramesh, Aditya et al. (2022). 'Hierarchical text-conditional image generation with clip latents'. In: *arXiv preprint arXiv:2204.06125* 1.2, p. 3.

Réau, Manon et al. (2018). 'Decoys selection in benchmarking datasets: overview and perspectives'. In: *Frontiers in Pharmacology* 9, p. 11.

Reed, Janet et al. (2023). 'AI image-generation as a teaching strategy in nursing education'. In: *Journal of Interactive Learning Research* 34.2, pp. 369–399.

Reymond, Jean-Louis (2015). 'The chemical space project'. In: *Accounts of Chemical Research* 48.3, pp. 722–730.

Riniker, Sereina and Gregory A. Landrum (2015). 'Better Informed Distance Geometry: Using What We Know To Improve Conformation Generation'. In: *Journal of Chemical Information and Modeling* 55.12. PMID: 26575315, pp. 2562–2574.

Robbins, Herbert and Sutton Monro (1951). 'A stochastic approximation method'. In: *The Annals of Mathematical Statistics*, pp. 400–407.

Rosenblatt, Frank (1958). 'The perceptron: a probabilistic model for information storage and organization in the brain.' In: *Psychological Review* 65.6, p. 386.

Röthlisberger, Daniela et al. (2008). 'Kemp elimination catalysts by computational enzyme design'. In: *Nature* 453.7192, pp. 190–195.

Rumelhart, David E, Geoffrey E Hinton and Ronald J Williams (1986). 'Learning representations by back-propagating errors'. In: *Nature* 323.6088, pp. 533–536.

Runcie, Nicholas T and Antonia SJS Mey (2023). 'SILVR: guided diffusion for molecule generation'. In: *Journal of Chemical Information and Modeling* 63.19, pp. 5996–6005.

Russell, Stuart J and Peter Norvig (2021). *Artificial Intelligence: A Modern Approach, Global Edition 4e*. Pearson.

Ruthotto, Lars and Eldad Haber (2021). 'An introduction to deep generative modeling'. In: *GAMM-Mitteilungen* 44.2, e202100008.

Saha, Chandra Nath and Sanjib Bhattacharya (2011). 'Intellectual property rights: An overview and implications in pharmaceutical industry'. In: *Journal of Advanced Pharmaceutical Technology & Research* 2.2, pp. 88–93.

Salentin, Sebastian et al. (2015). 'PLIP: fully automated protein–ligand interaction profiler'. In: *Nucleic Acids Research* 43.W1, W443–W447.

Sánchez-Cruz, Norberto (2023). 'Deep graph learning in molecular docking: Advances and opportunities'. In: *Artificial Intelligence in the Life Sciences* 3, p. 100062.

Santos, Emmanuel LC de los et al. (2016). 'Engineering transcriptional regulator effector specificity using computational design and in vitro rapid prototyping: developing a vanillin sensor'. In: *ACS Synthetic Biology* 5.4, pp. 287–295.

Santos, Gabriela B, A Ganesan and Flavio S Emery (2016). 'Oral administration of peptide-based drugs: beyond Lipinski's Rule'. In: *ChemMedChem* 11.20, pp. 2245–2251.

Scannell, Jack W et al. (2012). 'Diagnosing the decline in pharmaceutical R&D efficiency'. In: *Nature Reviews Drug Discovery* 11.3, pp. 191–200.

Scantlebury, Jack et al. (2023). 'A small step toward generalizability: training a machine learning scoring function for structure-based virtual screening'. In: *Journal of Chemical Information and Modeling* 63.10, pp. 2960–2974.

Scardino, Valeria, Juan I Di Filippo and Claudio N Cavasotto (2023). 'How good are AlphaFold models for docking-based virtual screening?' In: *iScience* 26.1.

Scavone, Cristina et al. (2019). 'The new paradigms in clinical research: from early access programs to the novel therapeutic approaches for unmet medical needs'. In: *Frontiers in Pharmacology* 10, p. 111.

Scheen, Jenke et al. (2025). 'Leveraging Alchemical Free Energy Calculations with Accurate Protein Structure Prediction'. In: *ChemRxiv*.

Schmirler, Robert, Michael Heinzinger and Burkhard Rost (2024). 'Fine-tuning protein language models boosts predictions across diverse tasks'. In: *Nature Communications* 15.1, p. 7407.

Schneuing, Arne et al. (2024). 'Structure-based drug design with equivariant diffusion models'. In: *Nature Computational Science* 4.12, pp. 899–909.

Schreiber, Stuart L (2021). 'The rise of molecular glues'. In: *Cell* 184.1, pp. 3–9.

Schrödinger, LLC (2015). 'The PyMOL Molecular Graphics System, Version 1.8'.

Segler, Marwin HS, Mike Preuss and Mark P Waller (2018). 'Planning chemical syntheses with deep neural networks and symbolic AI'. In: *Nature* 555.7698, pp. 604–610.

Sheils, Timothy K et al. (2021). 'TCRD and Pharos 2021: mining the human proteome for disease biology'. In: *Nucleic Acids Research* 49.D1, pp. D1334–D1346.

Shen, Chao et al. (2021). 'The impact of cross-docked poses on performance of machine learning classifier for protein–ligand binding pose prediction'. In: *Journal of Cheminformatics* 13 (1), pp. 1–18.

Shen, Chao et al. (2022). 'Boosting protein–ligand binding pose prediction and virtual screening based on residue–atom distance likelihood potential and graph transformer'. In: *Journal of Medicinal Chemistry* 65.15, pp. 10691–10706.

Shen, Chao et al. (2023). 'A generalized protein–ligand scoring framework with balanced scoring, docking, ranking and screening powers'. In: *Chemical Science* 14.30, pp. 8129–8146.

Shepherd, Claire et al. (2022). 'Surface plasmon resonance screening to identify active and selective adenosine receptor binding fragments'. In: *ACS Medicinal Chemistry Letters* 13.7, pp. 1172–1181.

Showalter, Scott A and Rafael Brüschweiler (2007). 'Validation of molecular dynamics simulations of biomolecules using NMR spin relaxation as benchmarks: application to the AMBER99SB force field'. In: *Journal of Chemical Theory and Computation* 3.3, pp. 961–975.

Sieg, Jochen, Florian Flachsenberg and Matthias Rarey (2019). 'In need of bias control: evaluating chemical data for machine learning in structure-based virtual screening'. In: *Journal of Chemical Information and Modeling* 59.3, pp. 947–961.

Silver, David et al. (2016). 'Mastering the game of Go with deep neural networks and tree search'. In: *Nature* 529.7587, pp. 484–489.

Simeon, Saw and Nathjanan Jongkon (2019). 'Construction of quantitative structure activity relationship (QSAR) Models to predict potency of structurally diversed janus kinase 2 inhibitors'. In: *Molecules* 24.23, p. 4393.

Singhal, Raghav et al. (2025). 'A general framework for inference-time scaling and steering of diffusion models'. In: *arXiv preprint arXiv:2501.06848*.

Sink, Roman et al. (2010). 'False positives in the early stages of drug discovery'. In: *Current Medicinal Chemistry* 17.34, pp. 4231–4255.

Škrinjar, Peter et al. (2025). 'Have protein-ligand co-folding methods moved beyond memorisation?' In: *BioRxiv*, pp. 2025–02.

Smith, Micholas Dean et al. (2024). 'Drugging the entire human proteome: Are we there yet?' In: *Drug Discovery Today* 29.3, p. 103891.

Song, Yang et al. (2020). 'Score-based generative modeling through stochastic differential equations'. In: *arXiv preprint arXiv:2011.13456*.

Sotriffer, Christoph A, Holger Gohlke and Gerhard Klebe (2002). 'Docking into knowledge-based potential fields: a comparative evaluation of DrugScore'. In: *Journal of Medicinal Chemistry* 45.10, pp. 1967–1970.

Speck-Planche, Alejandro and Valeria V Kleandrova (2022). 'Multi-condition QSAR model for the virtual design of chemicals with dual pan-antiviral and anti-cytokine storm profiles'. In: *ACS Omega* 7.36, pp. 32119–32130.

Spitzer, Russell and Ajay N Jain (2012). 'Surflex-Dock: Docking benchmarks and real-world application'. In: *Journal of Computer-Aided Molecular Design* 26.6, pp. 687–699.

Srinivasan, Jayashree et al. (1998). 'Continuum solvent studies of the stability of DNA, RNA, and phosphoramidate- DNA helices'. In: *Journal of the American Chemical Society* 120.37, pp. 9401–9409.

Stark, Hannes et al. (2023). 'Harmonic prior self-conditioned flow matching for multi-ligand docking and binding site design'. In: *NeurIPS 2023 AI for Science Workshop*.

Stärk, Hannes et al. (2022). 'Equibind: Geometric deep learning for drug binding structure prediction'. In: *International Conference on Machine Learning*. PMLR, pp. 20503–20521.

Steinbeck, Christoph et al. (2020). 'NFDI4Chem-towards a national research data infrastructure for chemistry in Germany'. In: *Research Ideas and Outcomes* 6, e55852.

Steinegger, Martin and Johannes Söding (2017). 'MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets'. In: *Nature Biotechnology* 35.11, pp. 1026–1028.

Stepniewska-Dziubinska, Marta M, Piotr Zielenkiewicz and Pawel Siedlecki (2018). 'Development and evaluation of a deep learning model for protein–ligand binding affinity prediction'. In: *Bioinformatics* 34.21, pp. 3666–3674.

Stokes, Jonathan M. et al. (2020). 'A Deep Learning Approach to Antibiotic Discovery'. In: *Cell* 180 (4), 688–702.e13.

Strieth-Kalthoff, Felix et al. (2022). 'Machine Learning for Chemical Reactivity: The Importance of Failed Experiments'. In: *Angewandte Chemie - International Edition* 61 (29).

Su, Minyi et al. (2018). 'Comparative assessment of scoring functions: the CASF-2016 update'. In: *Journal of Chemical Information and Modeling* 59.2, pp. 895–913.

Subramanian, Govindan et al. (2016). 'Computational modeling of β-secretase 1 (BACE-1) inhibitors using ligand based approaches'. In: *Journal of Chemical Information and Modeling* 56.10, pp. 1936–1949.

Sun, Chen et al. (2017). 'Revisiting unreasonable effectiveness of data in deep learning era'. In: *IEEE International Conference on Computer Vision*, pp. 843–852.

Sun, Duxin et al. (2022). 'Why 90% of clinical drug development fails and how to improve it?' In: *Acta Pharmaceutica Sinica B* 12.7, pp. 3049–3062.

Sun, Jiayu et al. (2018). 'Learning sparse representation with variational auto-encoder for anomaly detection'. In: *IEEE Access* 6, pp. 33353–33361.

Sutanto, Fandi, Markella Konstantinidou and Alexander Dömling (2020). 'Covalent inhibitors: a rational approach to drug discovery'. In: *RSC Medicinal Chemistry* 11.8, pp. 876–884.

Sutton, Richard S, Andrew G Barto et al. (1998). *Reinforcement learning: An introduction*. Vol. 1. 1. MIT press Cambridge.

Suzek, Baris E et al. (2007). 'UniRef: comprehensive and non-redundant UniProt reference clusters'. In: *Bioinformatics* 23.10, pp. 1282–1288.

Svetnik, Vladimir et al. (2003). 'Random forest: a classification and regression tool for compound classification and QSAR modeling'. In: *Journal of Chemical Information and Computer Sciences* 43.6, pp. 1947–1958.

Sykes, Richard A et al. (2024). 'What has scripting ever done for us? The CSD Python application programming interface (API)'. In: *Applied Crystallography* 57.4, pp. 1235–1250.

Tang, Jing et al. (2014). 'Making sense of large-scale kinase inhibitor bioactivity data sets: a comparative and integrative analysis'. In: *Journal of Chemical Information and Modeling* 54.3, pp. 735–743.

Thakkar, Amol et al. (2023). 'Unbiasing retrosynthesis language models with disconnection prompts'. In: *ACS Central Science* 9.7, pp. 1488–1498.

Torren-Peraire, Paula et al. (2024). 'Models matter: The impact of single-step retrosynthesis on synthesis planning'. In: *Digital Discovery* 3.3, pp. 558–572.

Touvron, Hugo et al. (2023). 'Llama 2: Open foundation and fine-tuned chat models'. In: *arXiv preprint arXiv:2307.09288*.

Tran-Nguyen, Viet-Khoa, Celien Jacquemard and Didier Rognan (2020). 'LIT-PCBA: an unbiased data set for machine learning and virtual screening'. In: *Journal of Chemical Information and Modeling* 60.9, pp. 4263–4273.

Tropsha, Alexander (2010). 'Best practices for QSAR model development, validation, and exploitation'. In: *Molecular Informatics* 29.6-7, pp. 476–488.

Tropsha, Alexander, Holli-Joi Martin and Artem Cherkasov (2025). 'The Six Ds of Exponentials and drug discovery: A path toward reversing Eroom's law'. In: *Drug Discovery Today*, p. 104341.

Trott, Oleg and Arthur J Olson (2010). 'AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading'. In: *Journal of Computational Chemistry* 31.2, pp. 455–461.

Turnbull, Andrew P and Paul Emsley (2013). 'Studying protein–ligand interactions using x-ray crystallography'. In: *Protein-Ligand Interactions: Methods and Applications*. Springer, pp. 457–477.

U.S. Food and Drug Administration (2018). *Step 3: Clinical Research*. `https://www.fda.gov/patients/drug-development-process/step-3-clinical-research`. Accessed: 2025-09-16.

— (2020). *Step 4: FDA Drug Review*. `https://www.fda.gov/patients/drug-development-process/step-4-fda-drug-review`. Accessed: 2025-09-16.

Umscheid, Craig A, David J Margolis and Craig E Grossman (2011). 'Key concepts of clinical trials: a narrative review'. In: *Postgraduate Medicine* 123.5, pp. 194–204.

Valsson, Ísak et al. (2025). 'Narrowing the gap between machine learning scoring functions and free energy perturbation using augmented data'. In: *Communications Chemistry* 8.1, p. 41.

Vamathevan, Jessica et al. (2019). 'Applications of machine learning in drug discovery and development'. In: *Nature Reviews Drug Discovery* 18.6, pp. 463–477.

Van Giffen, Benjamin, Dennis Herhausen and Tobias Fahse (2022). 'Overcoming the pitfalls and perils of algorithms: A classification of machine learning biases and mitigation methods'. In: *Journal of Business Research* 144, pp. 93–106.

Vapnik, Vladimir N (1999). 'An overview of statistical learning theory'. In: *IEEE Transactions on Neural Networks* 10.5, pp. 988–999.

Varadi, Mihaly et al. (2024). 'AlphaFold Protein Structure Database in 2024: providing structure coverage for over 214 million protein sequences'. In: *Nucleic Acids Research* 52.D1, pp. D368–D375.

Vaswani, Ashish et al. (2017). 'Attention is all you need'. In: *Advances in Neural Information Processing Systems* 30.

Veličković, Petar et al. (2017). 'Graph attention networks'. In: *arXiv preprint arXiv:1710.10903*.

Verdonk, Marcel L et al. (2003). 'Improved protein–ligand docking using GOLD'. In: *Proteins: Structure, Function, and Bioinformatics* 52.4, pp. 609–623.

Vilar, Santiago, Giorgio Cozza and Stefano Moro (2008). 'Medicinal chemistry and the molecular operating environment (MOE): application of QSAR and molecular docking to drug discovery'. In: *Current Topics in Medicinal Chemistry* 8.18, pp. 1555–1572.

Villar, Hugo O and Mark R Hansen (2009). 'Design of chemical libraries for screening'. In: *Expert Opinion on Drug Discovery* 4.12, pp. 1215–1220.

Vincent, Fabien et al. (2022). 'Phenotypic drug discovery: recent successes, lessons learned and new directions'. In: *Nature Reviews Drug Discovery* 21.12, pp. 899–914.

Virtanen, Pauli et al. (2020). 'SciPy 1.0: fundamental algorithms for scientific computing in Python'. In: *Nature Methods* 17.3, pp. 261–272.

Voitsitskyi, Taras et al. (2024). 'Augmenting a training dataset of the generative diffusion model for molecular docking with artificial binding pockets'. In: *RSC Advances* 14.2, pp. 1341–1353.

Volkov, Mikhail et al. (2022). 'On the Frustration to Predict Binding Affinities from Protein-Ligand Structures with Deep Neural Networks'. In: *Journal of Medicinal Chemistry* 65 (11), pp. 7946–7958.

Vondrick, Carl, Hamed Pirsiavash and Antonio Torralba (2016). 'Generating videos with scene dynamics'. In: *Advances in Neural Information Processing Systems* 29.

Wagle, Swapnil et al. (2023). 'Sunsetting binding MOAD with its last data update and the addition of 3D-ligand polypharmacology tools'. In: *Scientific Reports* 13.1, p. 3008.

Wallach, Izhar and Abraham Heifets (2018). 'Most ligand-based classification benchmarks reward memorization rather than generalization'. In: *Journal of Chemical Information and Modeling* 58.5, pp. 916–932.

Walters, Patrick (2023). *We Need Better Benchmarks for Machine Learning in Drug Discovery*. `https://practicalcheminformatics.blogspot.com/2023/08/we-need-better-benchmarks-for-machine.html`. Accessed: 2023-12-17.

— (2024). *Generative Molecular Design Isn't As Easy As People Make It Look*. `http://practicalcheminformatics.blogspot.com/2024/05/generative-molecular-design-isnt-as.html`. Accessed: 2025-04-05.

Wang, Cheng and Yingkai Zhang (2017). 'Improving Scoring-Docking-Screening Powers of Protein-Ligand Scoring Functions using Random Forest'. In: *Journal of Computational Chemistry* 38 (3), p. 169.

Wang, Chi et al. (2021a). 'FLAML: A Fast and Lightweight AutoML Library'. In: *Proceedings of Machine Learning and Systems*. Ed. by A. Smola, A. Dimakis and I. Stoica. Vol. 3, pp. 434–447.

Wang, Jian and Nikolay V Dokholyan (2019). 'MedusaDock 2.0: efficient and accurate protein–ligand docking with constraints'. In: *Journal of Chemical Information and Modeling* 59.6, pp. 2509–2515.

Wang, Renxiao et al. (2005). 'The PDBbind database: methodologies and updates'. In: *Journal of Medicinal Chemistry* 48.12, pp. 4111–4119.

Wang, Rongguang, Pratik Chaudhari and Christos Davatzikos (2023). 'Bias in machine learning models can be significantly mitigated by careful training: Evidence from neuroimaging studies'. In: *Proceedings of the National Academy of Sciences* 120.6, e2211613120.

Wang, Shuzhe et al. (2020). 'Improving conformer generation for small rings and macrocycles based on distance geometry and experimental torsional-angle preferences'. In: *Journal of Chemical Information and Modeling* 60.4, pp. 2044–2058.

Wang, Zechen et al. (2021b). 'OnionNet-2: A Convolutional Neural Network Model for Predicting Protein-Ligand Binding Affinity Based on Residue-Atom Contacting Shells'. In: *Frontiers in Chemistry* 9, p. 913.

Watson, Joseph L et al. (2023). 'De novo design of protein structure and function with RFdiffusion'. In: *Nature* 620.7976, pp. 1089–1100.

Wichard, Joerg D, Maciej J Ogorzałek and Christian Merkwirth (2015). 'CNN in drug design—Recent developments'. In: *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, pp. 405–408.

Wicky, Basile IM et al. (2022). 'Hallucinating symmetric protein assemblies'. In: *Science* 378.6615, pp. 56–61.

Wohlwend, Jeremy et al. (2025). 'Boltz-1 democratizing biomolecular interaction modeling'. In: *BioRxiv*, pp. 2024–11.

Wojcikowski, Maciej, Piotr Zielenkiewicz and Pawel Siedlecki (2015). 'Open Drug Discovery Toolkit (ODDT): a new open-source player in the drug discovery field'. In: *Journal of Cheminformatics* 7.1, pp. 1–6.

Wojcikowski, Maciej et al. (2019). 'Development of a protein–ligand extended connectivity (PLEC) fingerprint and its application for binding affinity predictions'. In: *Bioinformatics* 35.8, pp. 1334–1341.

Wójcikowski, Maciej, Pedro J Ballester and Pawel Siedlecki (2017). 'Performance of machine-learning scoring functions in structure-based virtual screening'. In: *Scientific Reports* 7.1, p. 46710.

Wong, Chi Heem, Kien Wei Siah and Andrew W Lo (2019). 'Estimation of clinical trial success rates and related parameters'. In: *Biostatistics* 20.2, pp. 273–286.

Wong, Felix et al. (2022). 'Benchmarking AlphaFold-enabled molecular docking predictions for antibiotic discovery'. In: *Molecular Systems Biology* 18.9, e11081.

Wong, Felix et al. (2023). 'Discovery of a structural class of antibiotics with explainable deep learning'. In: *Nature*, pp. 1–9.

World Health Organization (2024). *Global Health Estimates 2021: Leading causes of death*. `https://www.who.int/data/gho/data/themes/mortality-and-global-health-estimates/ghe-leading-causes-of-death`. Accessed: 2025-09-16. Geneva: World Health Organization.

Wouters, Olivier J, Martin McKee and Jeroen Luyten (2020). 'Estimated research and development investment needed to bring a new medicine to market, 2009-2018'. In: *JAMA* 323.9, pp. 844–853.

Wu, Zhenqin et al. (2018). 'MoleculeNet: a benchmark for molecular machine learning'. In: *Chemical Science* 9.2, pp. 513–530.

Wu, Zonghan et al. (2020). 'A comprehensive survey on graph neural networks'. In: *IEEE Transactions on Neural Networks and Learning Systems* 32.1, pp. 4–24.

Wuethrich, Kurt (1989). 'The development of nuclear magnetic resonance spectroscopy as a technique for protein structure determination'. In: *Accounts of Chemical Research* 22.1, pp. 36–44.

Xie, Yutong et al. (2022). 'How Much Space Has Been Explored? Measuring the Chemical Space Covered by Databases and Machine-Generated Molecules'. In: *The Eleventh International Conference on Learning Representations*.

Xu, Weidi et al. (2017). 'Variational autoencoder for semi-supervised text classification'. In: *AAAI Conference on Artificial Intelligence*. Vol. 31. 1.

Yerien, Damian E, Sergio Bonesi and Al Postigo (2016). 'Fluorination methods in drug discovery'. In: *Organic & Biomolecular Chemistry* 14.36, pp. 8398–8427.

Yildirim, Erdem (2022). 'Text-to-image generation AI in architecture'. In: *Art and Architecture: Theory, Practice and Experience*, p. 97.

Yim, Jason et al. (2023a). 'Fast protein backbone generation with SE (3) flow matching'. In: *arXiv preprint arXiv:2310.05297*.

Yim, Jason et al. (2023b). 'SE (3) diffusion model with application to protein backbone generation'. In: *arXiv preprint arXiv:2302.02277*.

Ying, Rex et al. (2018). 'Graph convolutional neural networks for web-scale recommender systems'. In: *24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 974–983.

Zdrazil, Barbara et al. (2024). 'The ChEMBL Database in 2023: a drug discovery platform spanning multiple bioactivity data types and time periods'. In: *Nucleic Acids Research* 52.D1, pp. D1180–D1192.

Zhang, Chao et al. (2024a). 'Targeting the undruggables—the power of protein degraders'. In: *Science Bulletin* 69.11, pp. 1776–1797.

Zhang, Donglu et al. (2012). 'Preclinical experimental models of drug metabolism and disposition in drug discovery and development'. In: *Acta Pharmaceutica Sinica B* 2.6, pp. 549–561.

Zhang, Yang and Jeffrey Skolnick (2004). 'Scoring function for automated assessment of protein structure template quality'. In: *Proteins: Structure, Function, and Bioinformatics* 57.4, pp. 702–710.

— (2005). 'TM-align: a protein structure alignment algorithm based on the TM-score'. In: *Nucleic Acids Research* 33.7, pp. 2302–2309.

Zhang, Zaixi et al. (2024b). 'PocketGen: Generating Full-Atom Ligand-Binding Protein Pockets'. In: *BioRxiv*, pp. 2024–02.

Zhao, Hongyu and Zongru Guo (2009). 'Medicinal chemistry strategies in follow-on drug discovery'. In: *Drug Discovery Today* 14.9-10, pp. 516–522.

Zhao, Yong and Michel F Sanner (2008). 'Protein–ligand docking with multiple flexible side chains'. In: *Journal of Computer-Aided Molecular Design* 22.9, pp. 673–679.

Zhavoronkov, Alex et al. (2019). 'Deep learning enables rapid identification of potent DDR1 kinase inhibitors'. In: *Nature Biotechnology* 37.9, pp. 1038–1040.

Zhou, Gengmo et al. (2023). 'Uni-mol: A universal 3d molecular representation learning framework'. In: *ChemRxiv*.

Zhou, Guangfeng et al. (2024). 'An artificial intelligence accelerated virtual screening platform for drug discovery'. In: *Nature Communications* 15.1, p. 7761.

Zhu, Hui, Jincai Yang and Niu Huang (2022). 'Assessment of the Generalization Abilities of Machine-Learning Scoring Functions for Structure-Based Virtual Screening'. In: *Journal of Chemical Information and Modeling* 62 (22), pp. 5485–5502.

Zhu, Hui et al. (2025). 'Augmented BindingNet dataset for enhanced ligand binding pose predictions using deep learning'. In: *NPJ Drug Discovery* 2.1, p. 1.

Zilian, David and Christoph A. Sotriffer (2013). 'SFCscore(RF): a random forest-based scoring function for improved affinity prediction of protein-ligand complexes'. In: *Journal of Chemical Information and Modeling* 53 (8), pp. 1923–1933.

Ziv, Yael et al. (2025). 'MolSnapper: conditioning diffusion for structure-based drug design'. In: *Journal of Chemical Information and Modeling* 65.9, pp. 4263–4273.

Zwanzig, Robert W (1954). 'High-temperature equation of state by a perturbation method. I. Nonpolar gases'. In: *The Journal of Chemical Physics* 22.8, pp. 1420–1426.