Fine-Tuning Methods for Predicting Protein-ligand Binding Affinity with Molecular LLMs

Joshua Fitch

Master of Science

School of Informatics

University of Edinburgh

2024

Abstract

Predicting binding affinity of a ligand and target protein is essential in applications like modelling biological systems and drug discovery. Due to the cost and difficulty of wet-lab experiments to determine strong target binding there is demand for computational methods that can cheaply, quickly and accurately predict binding affinity. Recently, protein and ligand large language models (LLMs) that provide information-rich embeddings of amino acid sequences and ligand SMILES strings have been used to achieve state-of-the-art performance in a variety of protein and ligand property prediction tasks. However, use of molecular LLMs in prediction of interaction properties is harder due to increased complexity and computational burden of having to model two molecules. This paper is the first to investigate how to best use molecular LLM models for the task of predicting binding affinity, and uses the PDBbind dataset of protein-ligand pairs. A variety of parameter-efficient fine-tuning (PEFT) methods common in NLP are assessed to overcome computational issues, with adding a joint multi-layer perceptron (MLP) to the final LLM layers having the best performance. Combining this with another effective fine-tuning method BitFit produces the state-of-art model for protein-ligand binding affinity prediction that only considers sequence level information, achieving an RMSE of 1.215 on the PDBbind 2016 core set and outperforming other methods by at least 7%. This is despite using smaller LMs and only having 3.5% of parameters as trainable. Results evidence the potential of using molecular LLMs and pre-training on unlabelled data to improve binding affinity prediction performance, overcoming a lack of high quality labelled data that limits current ML approaches. Additional results suggest performance could be improved even more given bigger molecular LMs and more pre-training on downstream tasks.

Research Ethics Approval

This project was planned in accordance with the Informatics Research Ethics policy. It did not involve any aspects that required approval from the Informatics Research Ethics committee.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Joshua Fitch*)

Acknowledgements

I would like to thank my supervisor Rik Sarkar for continued technical support and feedback throughout the project, as well as my family, my girlfriend Eleni and my housemate Will for supporting me personally as I did the project.

Table of Contents

1 Introduction **1**

1.1 Motivation 1

1.1.1 Importance of Predicting Binding Affinity 1

1.1.2 Benefit of Computational and ML Approaches 2

1.1.3 Deep Learning Methods to Model Binding Affinity 3

1.1.4 Challenges and Aims of Paper 4

1.2 Contribution 5

2 Background **7**

2.1 Biological Background 7

2.1.1 Proteins and Ligands 7

2.1.2 K_d, Ki and IC_{50} 8

2.2 Literature Review 9

2.2.1 Interaction-based and Non-interaction-based Methods 9

2.2.2 Previous Approaches 9

2.2.3 Limitations of Previous Methods and Molecular LLMs 10

2.2.4 Parameter-Efficient Fine-Tuning 11

2.2.5 Dataset and Our Models 12

3 Methodology **13**

3.1 Dataset 13

3.1.1 The PDBbind Database 13

3.1.2 Pre-processing 15

3.2 Baselines 15

3.2.1 Summary 15

3.2.2 Embedding Layers 16

3.2.3 Baselines 1 and 2 16

3.2.4 Language Modelling 17
3.2.5 Transformer Architecture 17
3.2.6 Baseline 3 18
3.3 Parameter-Efficient Fine-Tuning 19
3.3.1 Summary 19
3.3.2 Types of PEFT 20
3.3.3 Function Composition Methods 21
3.3.4 Parameter Composition Methods 21
3.3.5 Input Composition Methods 22
3.3.6 Combining Methods 23
3.4 Model Implementation Details 23

4 Results **24**
4.1 Summary 24
4.2 Comparison of Fine-Tuning Methods 25
4.2.1 Research Questions 25
4.2.2 Models and Experiments 25
4.2.3 Pre-trained LLMs vs End-to-end Encoding 26
4.2.4 Benefit of PEFT 27
4.2.5 Best Individual PEFT Approach 27
4.2.6 Performance of Our Combined Model 27
4.3 Comparison With Pre-existing Methods 28
4.3.1 Methods from Literature and Our Models 28
4.3.2 Comparison of Our Methods to Those in Literature 29
4.3.3 Comparison of Our Best Model to Interaction-based Methods 30
4.4 Effect of Protein Model Size 30
4.4.1 Different Model Sizes and Experiments 31
4.4.2 Effect of Model Size on Performance 31
4.5 Relative Importance of Protein and Ligand LLM 33
4.5.1 Using Only One of Protein or Ligand LM 33
4.5.2 Fine-tuning Only One of Protein or Ligand LM 33

5 Discussion **35**
5.1 Benefits of Using Molecular LLMs and PEFT 35
5.2 Best Methods and State-of-the-art Combined Model 36
5.3 Effect of Molecular LLM Characteristics 36

6 Conclusions **38**

Bibliography **40**

Chapter 1

Introduction

Predicting protein-ligand binding affinity is very important task due to its use in real world applications like drug discovery. ML methods have the potential to replace costly wet-lab experiments to determine binding strength but current methods are limited by lack of high quality data. This research project aims to improve performance of protein-ligand binding affinity prediction by taking advantage of protein and ligand LLMs pre-trained on un-labelled data. To overcome computation power and modelling issues associated with predicting an interaction property like binding affinity, various parameter-efficient fine-tuning (PEFT) methods and architectures are tested for protein-ligand binding affinity performance. These experiments result in a state-of-the-art protein-ligand binding affinity model from sequence level data, achieving an RMSE of 1.215 on the PDBbind 2016 core set, outperforming comparable methods by at least 7%. Therefore evidencing the efficacy of LM pre-training and PEFT in predicting interactive molecular properties and overcoming lack of high quality labelled data.

1.1 Motivation

1.1.1 Importance of Predicting Binding Affinity

Predicting the binding affinity of candidate proteins to a given target is a vital task in biological applications like the modelling of biological systems and drug discovery [1]. AI-assisted drug discovery in particular is an area that has achieved a lot of attention recently, as key goals within the drug discovery sector include providing vast speed-up and increased efficacy at every stage of the drug development pipeline [2]. One stage set to be revolutionised by AI is the identification of lead compounds that have the potential

to be new drugs [2], and predicting protein-ligand binidng binding affinity is vital in the identification of lead proteins [2]. Many drugs work by binding to and therefore inhibiting the action of target compounds known to be important in perpetuating disease states [3]. As such, predicting binding affinity to these targets can therefore identify drugs that are currently in use and may have potential for re-purposing, as well as validating the efficacy and potential of de novo designed drugs [1]. Computationally finding drugs to re-purpose is especially significant in the field of drug discovery, as re-purposed drugs have a minimised risk of failure, are cheaper, and their development is less time-consuming [4].

1.1.2 Benefit of Computational and ML Approaches

In previous research, binding affinity has been determined by wet-lab experiments that are often accurate yet slow and expensive, making them unsuitable to screen large amounts of proteins to find strong binders [5]. In recent years computational methods have been proposed to address some of these limitations, with the benefit of being able to screen large amounts of proteins much faster and cheaper than laboratory-based experiments [1]. These computational methods have the potential to improve the very high 90% failure rate of clinical drug development [6] by improving drug lead identification and target validation. Improvements in this area would also reduce the vast time burden and cost of drug development, estimated at around 1-2 billion dollars over 10-15 years [6], and therefore make the development of drugs more financially feasible.

Machine learning specifically is a key field of study where applications could rapidly increase efficiency and efficacy within pharmaceutical and biological settings including drug development. In particular, deep learning methods have shown promise in reducing the burden of drug development in a variety of contexts, including their deployment in the prediction of many protein and ligand properties including stability prediction, toxicity prediction and binding affinity [1]. Their ability to digest large amounts of unstructured data and identify complex patterns makes them a natural choice for predicting molecular properties like binding affinity [1]. How machine learning and deep learning can be used to accelerate drug development is shown in Fig. 1.1.

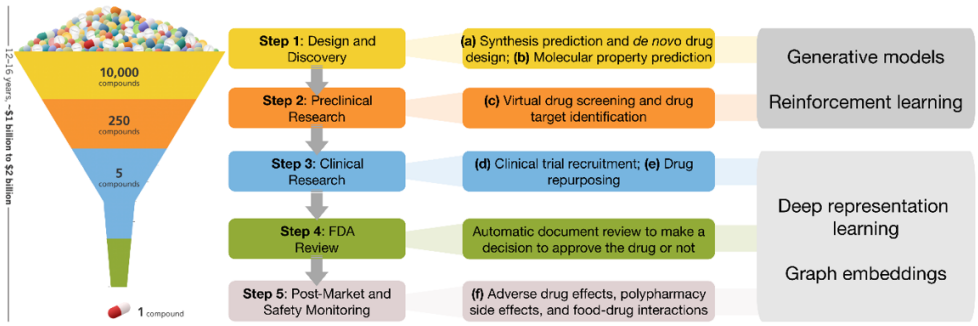

Figure 1.1: Summary of how Machine Learning methods can be used to reduce the cost and increase the efficacy of drug discovery. Image from https://zitniklab.hms.harvard.edu/drugl

1.1.3 Deep Learning Methods to Model Binding Affinity

Due to the importance of predicting protein-ligand binding affinity, many machine learning and non-machine learning approaches have been applied to the problem in the past. These have included physics based methods that simulate protein-ligand interactions to find the lowest entropy conformation [7], traditional machine learning (ML) methods that automatically learn from labelled data in a structured manner [8] and deep learning methods that capture patterns from labelled data in a more flexible way [9]. Deep learning methods have typically used convolutional neural network and graph neural network models for the task which can be effective but are limited by a lack of high-quality labelled data and are often reliant on higher order structural or interaction features that are not always available to researchers and expensive to obtain experimentally [1]. Therefore, this paper is the first to rigorously test the use of molecular language models (LMs) which take advantage of abundant unlabelled data for the binding affinity prediction task. Furthermore, models only use the primary structure of proteins and atomic structure of ligands to make predictions without the need for expensive structural or interaction features.

As protein molecules can be represented as a sequence of amino acids, and ligands as a sequence of atoms and bonds, with each variation of a sequence resulting in a different molecule, parallels between this and language formation can be drawn. Recent advancements in Natural Language Processing have led to the advent of protein LLMs that treat the primary structure of proteins - the amino acid sequence - as a sequence of words [10], along with ligand LLMs that treat the ligand SMILES (linearized atomic structure) as a sequence of words [11]. These models can be used to generate

embeddings of protein and ligand sequences which can be used in downstream tasks like property prediction. Deep learning models that utilise these embeddings are the new state-of-the-art for many property prediction tasks [10], [11].

1.1.4 Challenges and Aims of Paper

As molecular LLMs are a very recent development in the field, at present there has not been much experimentation on how to best use these models to maximize performance in downstream prediction tasks, especially interactive properties that require modelling two molecules [1]. The size of molecular LLMs means that training from scratch is not a feasible option for most groups due to limitations in computational power [12], and this issue is especially prevalent in binding affinity prediction where we have a protein and ligand sequence. As a result, the use of inexpensive fine-tuning techniques is essential in order to improve the feasibility of carrying out interactive regression tasks like binding affinity prediction using protein LLMs [13], and address the issue of current models being limited by a lack of high-quality data. This is a key gap within the existing field of research, as the versatility of utilising LMs in this context means that work focused on binding affinity prediction also has the potential to improve the prediction of many other interactive protein and ligand properties. These methods are also able to be adapted to the prediction of other non-interactive properties of molecules such as solubility and toxicity.

In order to address this current gap in research, this project will use computationally inexpensive PEFT methods, experimentally determining which methods and architectures work best on prediction of binding affinity using protein and ligand LLMs. Experiments are carried on the PDBbind dataset of protein kinases [14], with our methods compared to others in the literature to assess the efficacy of approaches used in this paper. This will allow us to determine whether pre-training on unlabelled data can learn useful features that allow improved binding affinity prediction when fine-tuned on limited labelled data, and whether PEFT methods can be used to achieve high performance with reasonable computation times. Further experiments are also carried out to look into some important design choices when building binding affinity prediction models using molecular LMs, like protein model size and type of LM pre-training. This helps practitioners focus time and computation power on approaches that will most improve performance.

1.2 Contribution

Improving cheap and time-efficient ML methods for binding-affinity prediction is important in applications like drug discovery, but current ML methods performance is limited by a lack of quality labelled data. This problem has been addressed in other molecular property prediction tasks by leveraging LMs pre-trained on large amounts of unlabelled data to learn important features and then fine-tuned for the downstream prediction tasks. These methods are challenging to apply for the task of binding affinity prediction however as it requires modelling two sequences, the protein and ligand, making fine-tuning very computationally intensive and making it hard to find the right architectural setup. To address this we test the ability of multiple model architectures and computationally inexpensive PEFT methods to improve the performance of binding affinity prediction. This way we can leverage feature extraction from molecular LMs whilst not creating models unfeasible for most practitioners to train. Overall, the main contributions of this paper are:

- First paper to thoroughly test the use of protein and ligand LLMs for prediction of protein-ligand binding affinity to overcome limitations imposed on current methods by lack of quality labelled data. PEFT is used to deal with increased computational burden of modelling two molecules
- Exploration of the best methods and architecture resulting in a state-of-the-art model for predicting binding affinity from sequence level information. This model uses BitFit to tune protein and ligand LMs, as well a joint non-linear MLP regression head for prediction. Achieves an RMSE of 1.215 on PDBbind core set 2016, a 7% improvement on the next best sequence based method
- Results evidence that pre-training on unlabelled data enables molecular LMs to learn features relevant for property prediction, enabling higher performance when fine-tuned with limited labelled data.
- PEFT is shown as a very effective and inexpensive way to tune molecular LMs, our state-of-the-art model has 3.5% of parameters as trainable
- Increased protein LM size improves performance but is far less efficient and effective than PEFT of a smaller LM like carried out in our method
- Embeddings from molecular LMs with additional property prediction pre-training are more informative and give better performance without fine-tuning. Fine-

tuning of LMs not already pre-trained on downstream tasks is more important for performance

Chapter 2

Background

2.1 Biological Background

The main basis of this research is built upon basic biological concepts. The following section aims to define key concepts such as protein structure, ligand structure, K_d, K_i, and IC_{50}.

2.1.1 Proteins and Ligands

The structure of a protein is determined by the sequence and number of amino acids which it is built from, with variations in sequences resulting in a chemically distinct protein with a unique three-dimensional structure, which may have a different function or specificity [15]. This base sequence of amino acids is known as the **primary structure** of a protein and is held together by peptide bonds that form between amino acids. Hydrogen bonds can then form between weak negatively charged nitrogen and oxygen atoms and weak positively charged hydrogen atoms, resulting in folded and helical protein structures known as the **secondary structure**. Further conformational change then results in additional bonds (hydrogen, disulphide, and ionic) forming between the side chains of a protein, causing the protein to change shape further and conform into the protein's **tertiary structure** which determines function and properties [15].

Ligands are small molecules that bind to another molecule resulting in the formation of a complex. The protein-ligand complex forms when a ligand binds to a specific site on the protein's tertiary structure, potentially causing conformational changes within the protein [16]. The atoms in a ligand and how they are connected are referred to as its

atomic structure which determines ligand properties. The atomic structure of a ligand can be represented in a string form by ligand SMILES (Simplified Molecular Input Line Entry System). The ligand atomic structure or SMILES and the tertiary structure of the protein (that forms as a result of its primary structure) determines protein-ligand binding affinity [16].

2.1.2 K_d, Ki and IC_{50}

Binding affinity is the strength of the interaction between binding molecules. There are several ways in which the binding affinity of a ligand (A) can be defined - directly and indirectly. It can be translated into physicochemical terms directly as a dissociation constant (K_d), which is a measurement of how tightly a ligand binds to a receptor [17]. The equation representing K_d when a system is in equilibrium is as follows:

$$K_d = \frac{[A][B]}{AB} \tag{2.1}$$

Where [A] and [B] represent concentrations of the ligands and receptors respectively (the reactants), and AB represents the concentration of the bound complex (the product).

Indirectly, ligand binding affinity can be determined as an IC_{50} value, using a competition binding experiment which determines the concentration of a ligand required to displace 50% of a fixed concentration of reference ligand [17].

The affinity of the receptor (A) to bind with a ligand (B) is represented by K_i, an inhibition constant that denotes the concentration required to occupy 50% of the receptor [17].

The relationship between all three binding affinity constants is summarised in the below equation:

$$K_i = \frac{IC_{50}}{1 + [L_t]/K_d} \tag{2.2}$$

Where $[L_t]$ represents the concentration of a labelled ligand, and K_d and IC_{50} are as defined above.

2.2 Literature Review

2.2.1 Interaction-based and Non-interaction-based Methods

In the past, a variety of methods have been used to carry out the prediction of protein-ligand binding affinity, and these methods can be broadly categorised into two groups - interaction-based and non-interaction-based [18]. Interaction-based methods include data pertaining to how the target protein and ligand interact, whereas non-interaction-based methods exclusively have features from the individual proteins and ligands [18]. Within these broad themes, methods can be further categorised based on whether experimental 3D structural features are used as input to models or not [1]. This project focuses on non-interaction-based methods that only use the primary structure (the amino acid sequence) of the protein and the ligand SMILES as an input to the model. We have chosen to do this as these models are more widely applicable since both interactions and structural features are not always available in a real-world setting and are expensive to find experimentally [1].

2.2.2 Previous Approaches

The earliest computational methods to predict protein-ligand binding affinity were physics-based methods, using statistical mechanics and molecular dynamics simulations to estimate the conformation dynamics of the ligand and receptor [19]. Open-source programs such as Autodock Vina predict the non-covalent binding of receptors and ligands using a gradient optimisation method to predict molecular docking and virtual screening, these physics based methods find the lowest entropy conformation of protein and ligand binding [7]. After these physics-based methods, many traditional machine learning(ML) methods were used to improve upon the binding affinity prediction of protein and ligand pairs, for example multiple regression [20] and support vector regression [21]. One notable example is Fandom Forest (RF)-Score, that uses interaction features (based on proximity) and the random forest ML algorithm to implicitly capture binding effects that are harder to model explicitly, allowing problematic modelling assumptions used in physics-based methods to be circumvented [8]. RF-Score v3 is an updated model that uses an enhanced set of features and more diverse training data [22].

In recent research, deep learning methods have far outnumbered traditional ML and physics-based approaches to the binding affinity prediction problem and have since achieved much better performance [18]. Notable examples are InteractionGraphNet

(IGN) that uses a graph neural network [9], DeepDTA that uses a convolutional neural network [23], and CAPLA that uses an attention based approach [24]; all three methods use deep learning to sequentially learn the intramolecular and intermolecular interactions between proteins and ligands. The main ways these models differ pertain to both the types of architecture and each study's respective considerations of mutual interaction features. IGN uses a molecular graph representation of the 3D structures of protein complexes to predict binding affinity, using two stacked independent graph convolution modules [9]. This differs from the DeepDTA approach, which exclusively uses 1D representations of proteins and ligands (protein sequences and SMILES strings) within their convolution blocks, combining representations and feeding them into a three layer MLP regression head [23]. Alternatively, CAPLA uses a binding pocket input representation and cross-attention mechanism to explicitly model the interaction features between proteins and ligands. Within this method, dilated convolutions learn long-range features, and the model uses a feed forward network for prediction. There are two versions of CAPLA, the default model takes advantage of structural features, whilst CAPLA-Pred only uses sequence level information [24].

2.2.3 Limitations of Previous Methods and Molecular LLMs

Until recently, deep learning models for protein and ligand property prediction, including protein-ligand binding affinity prediction were typically trained end-to-end or using hand-crafted feature extraction methods, and used convolutional or graph neural network architectures [18], [25], [26]. These methods achieved some promising results, however, they were hindered by a lack of high-quality data, often struggling to generalise to molecules that were dissimilar to those in the training set [27]. Since then, the state-of-the-art in many molecular property prediction tasks has been improved by taking advantage of the recent development of protein and ligand LLMs [12]. These are transformer-based models inspired by advances in natural language processing, and trained using a masked prediction objective on large volumes of unlabelled amino acid sequences or ligand SMILES instead of text, with notable examples including ESM-2 [10], ChemBERTa-2 [11] and ProteinBERT [28]. These models can be used to obtain information-rich embeddings of both ligands and proteins to help predict a variety of molecular properties including protein-ligand binding affinity. An example of a model built on top of these embeddings can be seen in Fig 2.1.

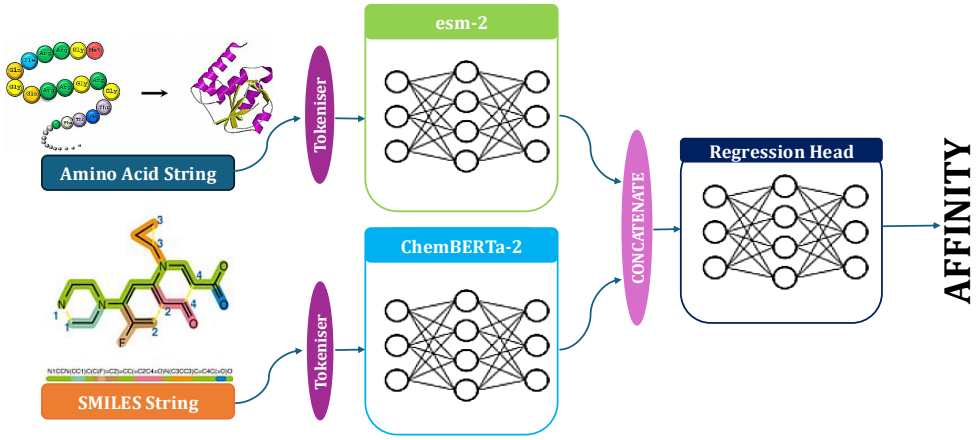

Figure 2.1: Example of how embeddings from protein LLMs can be used to carry out binding affinity prediction on the protein amino acid sequence and ligand SMILES string

2.2.4 Parameter-Efficient Fine-Tuning

Due to the recent adoption of protein and ligand LLMs for use in predicting molecular properties, and the lack of research into using molecular LLMs to predict binding affinity, it is not yet clear how to best utilise protein and ligand LLMs to carry out this task. Most bodies of research into molecular property prediction with LLMs so far have focused on building network architectures on top of embeddings from protein and/or ligand LLMs, using pre-trained, frozen LLMs to encode sequences [11]. This is mainly due to the large size of modern LMs, meaning full fine-tuning is often not possible [29]. However, in the field of NLP there are a variety of parameter efficient methods used to very effectively and cheaply adapt LLMs to downstream classification or regression tasks that could apply to binding affinity prediction [29]. Common approaches include: 1) adding trainable functions to frozen LMs, such as regression heads or adapters at the end of each block (function composition) [30], [31]; 2) updating only specific parameter groups, like bias weights or low-rank weights (parameter composition) [32], [33]; and 3) inserting trainable tokens into sequences, for example, by pre-pending trainable embeddings (input composition) [34].

2.2.5 Dataset and Our Models

This paper tests the ability of PEFT methods and molecular LLMs to improve the prediction of protein-ligand binding affinity by leveraging pre-training on unlabelled data and inexpensive fine-tuning methods. This addresses limitations of current methods created by lack of high-quality labelled data [1], and the computational burden of fully fine-tuning multple LLMs for interactive property prediction. Models were trained and tested on the widely used [1] PDBbind dataset of experimentally validated protein-ligand binding affinities from the Protein Data Bank (PDB) [14]. The PDBbind 2016 core set is used for testing as its wide use in literature makes comparison with other methods more reliable [1]. Protein amino acid sequences and ligand SMILES strings extracted from the PDBbind dataset are the only input to models we create in this paper.

Chapter 3

Methodology

Experiments within this paper are carried out on the PDBbind dataset [14], testing a variety of approaches to parameter-efficient fine-tuning protein LLMs for use in binding affinity prediction.

3.1 Dataset

3.1.1 The PDBbind Database

The PDBbind (Protein Data Bank bind) dataset [14] is a set of experimentally validated binding affinities for protein-ligand complexes taken from the Protein Data Bank. The dataset consists of 23,496 total entries, including 19,443 protein-ligand entries [14]. PDBbind has been chosen for use in this paper due to the relatively large amount of protein-ligand entries it contains with experimentally validated binding affinity compared to other binding-affinity datasets, which has the dual effect of making training a high-performing model easier and making validation more reliable [35]. Furthermore, PDBbind is the most commonly used dataset for predicting binding affinity [36], so using PDBbind ensures that the efficacy of models built in this paper can be compared fairly with a variety of other methods in the existing literature. There are multiple versions of the PDBbind database as it is consistently updated. This paper uses the 2020 version of the database for training, as this is the most recent version that can be obtained without a subscription [14], [37]. We use a subset of the 2016 version for testing as this is the most commonly used benchmark in the literature, and so is the best for comparison to other methods [36].

Protein-ligand binding data points in the PDBbind dataset contain information on

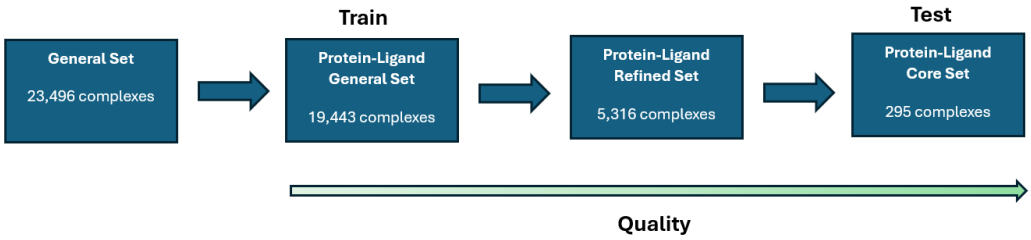

Figure 3.1: Summary of the hierarchy of sets in the PDBbind dataset [14]. This paper uses the general set (excluding the core set) for training and the core set for testing

the protein and ligand individually, as well as information on the interactions between the protein and ligand. This includes information such as the primary, secondary, and tertiary structure of proteins; amino acid spatial position; atom and bond information of ligands; and structural information on the protein-ligand pocket, all of which can be potentially used to predict binding affinity [14]. This paper only uses two features to predict binding affinity - the protein's primary structure (or amino acid sequence) and the chemical structure of the ligand. This is because both higher-order information about the 3-dimensional structure of proteins and information about protein-ligand interaction may not be readily available, for example in the case of protein discovery via genomics [38]. Additionally, this data is time-consuming and expensive to obtain experimentally [5]. A model built to predict binding affinity based purely on the amino acid sequence of proteins and chemical structure of ligands is therefore applicable to more real-world scenarios [21].

Binding affinity data is in the form of either k_d, k_i, or IC_{50} values as explained in the Biological Background section. The dataset is split into a hierarchy based on the quality of the protein-ligand complexes, summarised in Fig. 3.1:

- General set: contains all protein-ligand complexes
- Refined set: a higher quality subset of the general set filtered using binding data, crystal structures, and the nature of complexes
- Core set: a subset of the refined set with even higher quality, therefore this set is used to validate AI models in this paper and many others in the literature [14], [36]

3.1.2 Pre-processing

Protein data is in the form of .pbd files, from which the amino acid sequence of the protein is extracted using Biopython [39]. Ligand data is contained within .sdf files, with the atomic structure extracted using rdkit [40]. The graph form atomic structure is then converted to a linearised representation so that it can be input into a language model (LM). Binding affinity values are downloaded and matched with corresponding protein-ligand complexes using their IDs. A small number of invalid .pdb or .sdf files that could not be parsed were removed from the dataset, and proteins longer than 1024 amino acids were discarded to speed up the runtime of protein LMs. After the core set complexes were removed from the general set to avoid train-test overlap, this finally left 19,134 protein-ligand complexes in the general set for training, with 290 protein-ligand complexes present in the core set for testing. Tokenizers downloaded with the ESM-2 protein LM [10] and ChemBERTa-2 ligand LM [11] were used to tokenize protein and ligand sequences respectively before being input to the LMs. The resulting tokens are referred to as the vocabulary of the LMs, and going forward we denote the length of these vocabularies as V.

3.2 Baselines

3.2.1 Summary

Various baselines were used to ensure the efficacy of methods, including simple embedding models and regression models using LMs as encoders. All models take in a sequence of tokens derived from protein amino acids and ligand SMILES, and output a prediction for the binding affinity between that protein and ligand. A summary of the baselines is below:

1. Simple Embedding + Linear Regression: Protein and ligand are embedded and these embeddings are concatenated before being fed to a linear regression layer. The model is trained end-to-end.

2. Simple Embedding + MLP: Protein and ligand are embedded and these embeddings are concatenated before being fed to a 2-layer non-linear MLP. The model is trained end-to-end. Baselines 1 and 2 provide a comparison of language model embeddings to those trained from scratch.

3. LM Embedding + Linear Regression: Protein and ligand are embedded using separate LMs before being passed to a linear regression layer. Only the linear regression layer is trained. This baseline provides a comparison to test whether parameter-efficient fine-tuning (PEFT) methods from section 3.3 improve how informative LM embeddings are for binding affinity prediction.

More details on the workings of these baselines are given below.

3.2.2 Embedding Layers

To convert categorical tokens to vectors that can be processed by networks, each model starts with a separate embedding layer for the protein and ligand. The embedding layer takes in a token ID in the form of a one-hot encoded vector length V. This vector indexes a row in a large embedding matrix $E \in R^{V \text{ x } d}$ corresponding to a vector representation of that token, which has length d - the embedding size of the model. The embedding process is summarised in equation 3.1.

$$y = E^T \cdot x \tag{3.1}$$

Where x is the one-hot encoded ID vector and y is the output embedding. In the simple baselines 1 and 2, these embedding layers are trained from random initialisation, whereas in baseline 3 the embedding layers as part of protein and ligand LMs have already been trained and so are frozen at train time. Values of the embedding size d are kept similar between models for comparison, with these values being set to 350 for proteins and ligands in baselines 1 and 2, and 320 for proteins, 384 for ligands in baseline 3.

3.2.3 Baselines 1 and 2

After this, in baselines 1 and 2, the embedding layer vectors for tokens are aggregated by averaging. In baseline 1 these average representations are then processed by a linear regression model to produce a binding affinity value, whereas in baseline 2 they are processed by a 2-layer MLP. The MLP uses a 512 hidden dimension, picked based on a commonly used rule from [41], ReLU activation for non-linearity, and dropout with a probability of 0.2 to prevent over-fitting [42].

3.2.4 Language Modelling

In baseline 3, protein and ligand embeddings are processed further by two transformer LMs: protein LM ESM-2 [10] and ligand LM ChemBERTa-2 [11]. ChemBERTa-2 is a variant of the RoBERTa language model [43] that uses the BERT architecture [44] trained on chemical SMILES strings. ESM-2 has a similar architecture to BERT and is trained on protein sequences. Both models are pre-trained using masked language modelling, where a percentage of tokens are replaced with a mask token that the model is required to predict [44]. By learning to predict masked tokens, models are forced to learn complex embeddings of tokens and molecules that contain information on structure, properties, and sites of importance [10], [11]. These information-rich embeddings can then be used for downstream tasks. ChemBERTa-2 carries out additional pre-training by adding a regression head on top of embeddings for multiple downstream regression tasks, whereas ESM-2 is only trained using masked language modelling [10], [11]. Note that there are multiple pre-trained ESM-2 models of different sizes, we choose to use the smallest model with 8 million parameters in our experiments to reduce computational burden [10].

3.2.5 Transformer Architecture

The transformer architecture is composed of multiple stacked transformer blocks [45], of which there are three in ChemBERTa-2 and six in the small version of ESM-2 used in this paper. Transformer blocks in both ChemBERTa-2 and ESM-2 are made up of a self-attention layer, layer normalisation, projection to embedding dimension, and then a final layer normalisation [10], [11]. In ChemBERTa-2, skip connections are between the input and first layer normalisation, and between the first layer normalisation and second layer normalisation, whereas in ESM-2 skip connections are from the projection layer in the previous block to after the attention layer, and from after the attention layer to after the projection layer. ESM-2 uses a two-layer non-linear MLP as the projection layer whereas ChemBERTa-2 exclusively uses a linear layer. The most important part of the transformer block is the self-attention mechanism, originally inspired by alignment for translation [45]. The attention mechanism computes a dot product between every token embedding (key) and every other token embedding (query) to find a set of importance values between all tokens. Importance values are then converted to probabilities by softmax and used to compute a weighted sum of tokens which becomes the new embedding for that token. In practice, each token embedding is

converted to a query, key, and value via a linear projection to increase the expressiveness of the attention mechanism [45] as shown in equation 3.2.

$$Q = XW_q, \quad K = XW_k, \quad V = XW_v \tag{3.2}$$

Where X is the input matrix with each row representing the embedding for each token in the sequence and W_q, W_k and W_v are trainable weight matrices. Each row of Q, K and V represents the query, key, and value for that token. The attention mechanism is then carried out using equation 3.3.

$$Y = softmax(\frac{QK^T}{\sqrt{d}})V \tag{3.3}$$

Where QK^T is calculating the dot product between every query and key and $\sqrt{d}$ is the root of the embedding dimension used as scaling to maintain stable values and gradients [45]. Probabilities are multiplied by value vectors to get a matrix of new embeddings Y. Both ChemBERTa-2 and ESM-2 use narrow multi-head self-attention, in which token embeddings are split into parts based on the number of heads. Self-attention is then computed with separate weight matrices W_q, W_k, W_v on each part of the token, and then the final output embedding of dimension d for each token is formed by the concatenation of the output of each head [45]. Note that the dot product and the attention mechanism are permutation equivariant - the output embedding of a token would be the same regardless of its position in the input sequence [45]. However, this is not reflective of biology, where the exact position of an amino acid or atom is important in determining the properties of a molecule [15]. To rectify this, positional embeddings calculated from functions that map positions to real-valued vectors are added to tokens before they are input into the first transformer block. This way models are given positional information about each element in the sequence [45].

3.2.6 Baseline 3

As ChemBERTa-2 and ESM-2 are encoder models, the output of the last transformer block for both is a set of embeddings corresponding to tokens in the input sequence. However, how these embeddings are used to make binding affinity predictions is different between models. ChemBERTa-2 uses the embedding from a special [CLS] token concatenated to the start of the input sequence. This token is specifically added to capture sequence-level information that can be used for downstream property prediction [43]. ESM-2 doesn't use a [CLS] token and instead averages the embeddings from all

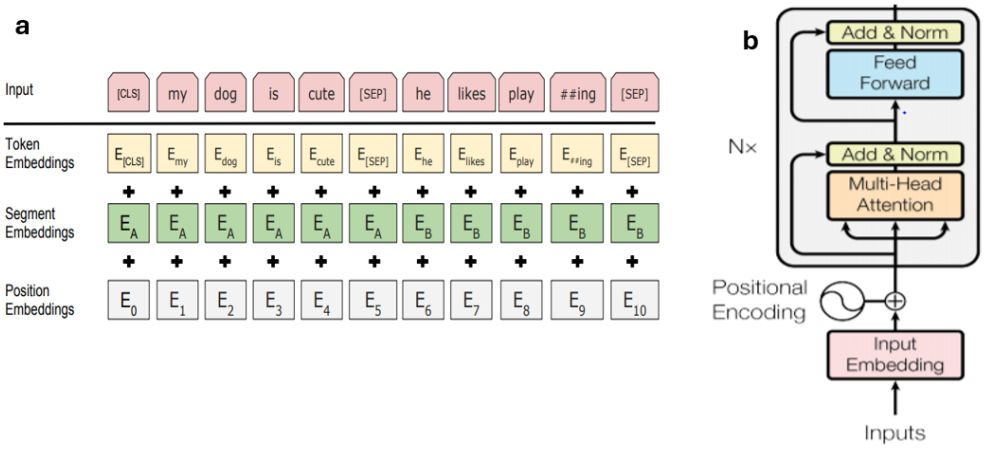

Figure 3.2: **a)** An example of how an input sequence is converted to embeddings that can be passed into transformer blocks in ChemBERTa-2. The process is the same for ESM-2 but without the [CLS] token. Note that the segment embeddings are identical for all tokens in our experiments, as only one segment is used for each LM. Image from [44]. **b)** Structure of a transformer block in ChemBERTa-2. The transformer block in ESM-2 is identical other than the positions of skip connections as described in 3.2.5. Image from [45].

tokens in the input sequence, and this average embedding is used for binding affinity prediction [10]. A summary of the input representation for ChemBERTa-2 and the transformer block can be seen in fig. 3.2.

3.3 Parameter-Efficient Fine-Tuning

3.3.1 Summary

We tested various commonly used and effective parameter-efficient fine-tuning (PEFT) methods for protein and ligand LLMs, intending to improve performance in the binding affinity prediction task. These included function composition, parameter composition, and input composition approaches. A summary of tested methods is below:

1. Joint MLP Adapter: Protein and ligand embeddings are concatenated and fed into a trainable two-layer non-linear MLP.

2. All Layer Adapter Tuning: Trainable two-layer non-linear MLP and layer normalisation are added to each transformer block in LMs, with embeddings concatenated and the trainable linear regression layer used for binding affinity prediction.

3. BiasFit: All bias weights in LMs are left as trainable, with concatenation and a linear regression layer used for prediction.

4. Low-Rank Adaptation: Trainable low-rank matrices are added to the self-attention layer in transformer blocks. Concatenation and a linear regression layer are used for prediction.

5. Prefix Tuning: Trainable prefix embeddings are concatenated to key and value matrices in every layer of LMs. Concatenation and a linear regression layer are used for prediction.

More detail on methods and implementation is given below.

3.3.2 Types of PEFT

The fine-tuning methods tested for protein-ligand binding affinity can be categorised into three groups depending on what kind of composition is used between frozen weights and trainable weights. Function composition involves adding new task-specific weights to augment a model's function $g(x) = f_\theta \odot f_\phi(x)$ where $g(x)$ is the output of a layer or layers in the network given input x. The function f_θ with parameters θ is frozen, while the function f_ϕ with parameters ϕ is trainable. $\odot$ represents composition of these functions [30], [46] . Parameter composition only updates weights in a specific group according to inductive biases about finding high-performing weights. The equation for parameter composition is $g(x) = f_{\theta \oplus \phi}(x)$ where $\oplus$ represents an update of a subset ϕ of all parameters θ, which could be implemented by updating parameters directly or adding trainable parameters to selected frozen parameters [32], [33]. Input composition augments the input of a model or layer with a trainable vector ϕ, which can be seen in the equation $g(x) = f_\theta([x, \phi])$ where ϕ represents the additional trainable parameters, and $[\cdot, \cdot]$ represents concatenation [34]. Note that we leave layer normalisation as trainable for all methods and all layers, as this stabilises training and increases adaptation capacity whilst adding very few trainable parameters [47].

3.3.3 Function Composition Methods

We tested two fine-tuning methods that use function composition. Firstly we tested adding a two-layer MLP with ReLU activation to the concatenated embeddings from ESM-2 and ChemBERTa-2. This can be thought of as adding a joint non-linear adapter to the last transformer block of the protein and ligand LMs where only this adapter is trainable, a common way of fine-tuning deep learning models [46]. This method tests the impact of both exclusively fine-tuning an adapter to the last layer of LMs, and also the effectiveness of jointly modelling protein and ligand embeddings with a non-linear function. The MLP has a 512 dimension hidden layer picked according to [41], and a dropout probability of 0.2 to prevent overfitting [42].

Secondly, we tested adapter tuning, which adds trainable modules to each transformer block in the protein and ligand LM. Adapter tuning was introduced in [30], but we used the version from [31] that only adds trainable adapters onto the end of each transformer block, as it is shown to be more efficient and has similar performance levels [33]. Adapters include: 1. a non-linear MLP with a hidden layer size smaller than the LM embedding dimension; 2. a skip connection from before the final layer norm in the original transformer block to after the adapter MLP; 3. a final layer norm. We used a hidden layer size of half the LM embedding dimension for the adapter MLP based on results in [48] and ReLU activation to introduce non-linearity. Embeddings from the last transformer block in each LM are converted to a binding affinity prediction by concatenation and a trainable linear regression layer.

3.3.4 Parameter Composition Methods

We also tested two parameter composition methods, BitFit and Low-rank adaptation (LoRA). Bitfit is a simple method that works by only updating bias weights in the protein and ligand LMs, and leaving all other weights frozen [32]. The final layer embeddings of both LMs are then concatenated before a linear regression layer is used to make the binding affinity prediction.

LoRA works by taking advantage of the inductive bias that an effective model for a new task can be found by only updating weights in a low-dimensional, randomly oriented subspace of the original weight space [33]. For each weight matrix W of shape m x n that is updated, LoRA uses two new trainable matrices A and B, with shape r x n and m x r respectively. By multiplying trainable matrices A and B together, and then adding them to the original matrix W, this is the equivalent of only updating a subset

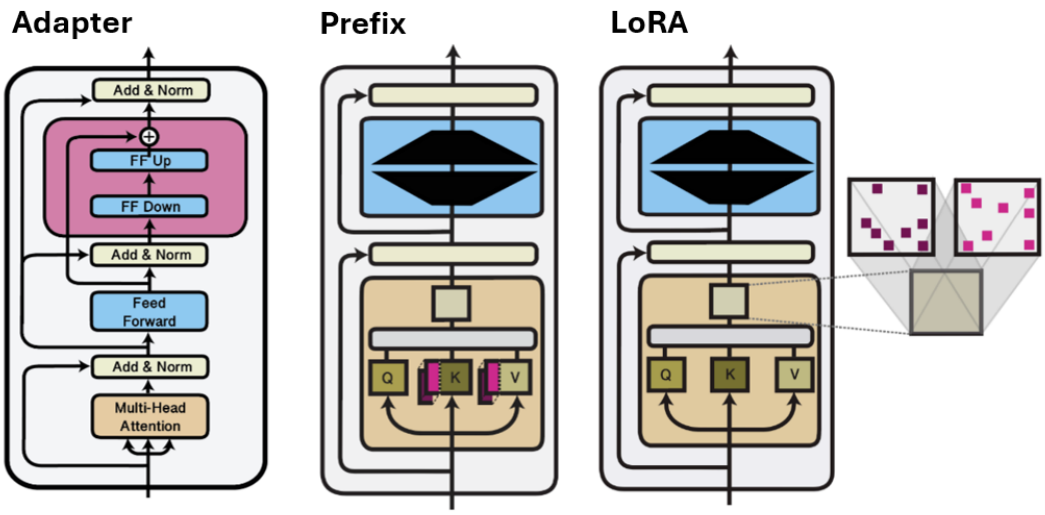

Figure 3.3: Graphical Illustration of three types of fine-tuning used in this paper: adapter tuning, prefix tuning and LoRA [30], [34], [33]. The adapter tuning figure shows the setup for tuning ChemBERTa-2. The setup for ESM-2 is similar but with the skip connections in different places as described in section 2.2. Note also in our implementation LoRA is only applied to query, key, and value weights and not to the projection layer.

of the weights of W that lie in a low dimensional subspace [33]. As high performance can be achieved by using LoRA to adapt weights in the attention layer of LMs [49] we add separate A and B matrices to the query, key, and value weights in every transformer block of the protein and ligand LMs as shown in equation 2.4.

$$W_k = W_k + \frac{\alpha}{r}(B_k A_k), \quad W_q = W_q + \frac{\alpha}{r}(B_q A_q), \quad W_v = W_v + \frac{\alpha}{r}(B_v A_v) \tag{3.4}$$

Where original attention weights W are frozen and added matrices A and B are trainable. We chose a rank r of 32 and scaling factor α (which controls the size of the update) of 64 according to results from [33]. Concatenation of protein and ligand embeddings and a linear regression layer are used to make the final prediction.

3.3.5 Input Composition Methods

Finally, we tested an input composition method, prefix tuning. Prefix tuning works by directly learning a continuous trainable prompt that is pre-pended to the input of a model or layer [34]. We used the setting in [34] that implements prefix tuning by concatenating a set of continuous embeddings, each the size of the embedding dimension d, to the

key and value matrices in the attention mechanism of every transformer block. By only adding to the key and value, the trainable prefix embeddings influence the attention mechanism, but do not increase the number of output token embeddings [34]. We used 200 trainable prefix embeddings in each block due to this having the highest performance in results from [34]. Binding affinity is then predicted by concatenation of protein and ligand embeddings followed by a linear regression layer. A graphical summary of adapter tuning, LoRA and prefix tuning can be seen in Fig. 3.3.

3.3.6 Combining Methods

Many of the fine-tuning methods we tested can be carried out at the same time to further increase the number of trainable parameters and potentially also model performance. To investigate this we also ran tests combining some of the fine-tuning methods described here. Due to promising results in initial tests, we further analysed the effectiveness of combining a non-linear MLP regression head (fine-tuning method 1), with updating the bias weights of LMs in BitFit (fine-tuning method 3).

3.4 Model Implementation Details

All models were implemented in Python using the PyTorch framework [50]. Models were trained with a batch size of 1024 to balance the speed of training and robustness of gradient updates [51], and where GPU memory did not allow this gradients were accumulated till 1024 samples had been processed, keeping comparison between models fair. Based on commonly used values, a learning rate of 0.001 was used for parameters trained from scratch like regression heads, and a learning rate of 0.00001 was used for fine-tuning pre-trained weights within the LMs. To aid convergence, the AdamW optimiser [52] and plateau learning rate scheduler were used. If models went 10 epochs without improvement in validation metrics then the learning rate was reduced by a factor of 10, and if 20 epochs passed without improvement then training was stopped and the highest performing model of the run was saved. All models were trained using mean squared error (MSE) loss, with the evaluation metric being root mean squared error (RMSE). Fine-tuning methods were implemented by adapting code from the github repository (link) for ESM-2 [10] and from the Hugging Face transformers github (link) [53] for ChemBERTa-2 [11].

Chapter 4

Results

4.1 Summary

This chapter details the results of tests using the algorithms described in the methodology. Throughout the results commonly used evaluation metric root mean squared error (RMSE) on the PDBbind 2016 core set [14] is used to assess and compare the performance of different models. The main results are:

- Encoding with pre-trained LLMs is shown to be more effective than encoding with models trained end-to-end.
- Parameter-efficient fine-tuning (PEFT) of molecular LMs produces much better performance than just using frozen LMs for binding affinity prediction
- Adding a joint MLP regression head is the most effective individual fine-tuning method with an RMSE of 1.313 and 3.2% trainable parameters.
- Combining BitFit with a joint MLP regression head gives a state-of-the-art sequence level binding affinity prediction model, achieving an RMSE of 1.215, 0.83 (7%) better than any comparable method and with 3.5% trainable parameters.
- Increasing protein LM size considerably increases performance of models with linear regression heads and slightly increases performance of models with non-linear MLP regression heads. Increasing protein LM size is not as effective or parameter-efficient as PEFT applied to a smaller model.
- Models using just the frozen ChemBERTa-2 ligand LM which has additional property pre-training outperform models using just the frozen ESM-2 protein model. PEFT on ESM-2 results in more performance gains than PEFT on ChemBERTa-2.

4.2 Comparison of Fine-Tuning Methods

This section presents results comparing the performance of baseline methods described in methodology section 3.2 and fine-tuning methods described in section 3.3.

4.2.1 Research Questions

Firstly, we compare results of baselines and fine-tuning methods to answer three main questions:

1. Is encoding with pre-trained LLMs more effective than encoding with models trained from scratch?
2. Is parameter-efficient fine-tuning (PEFT) more effective than just encoding with pre-trained LLMs and adding a simple regression head?
3. Which PEFT methods are most effective?

4.2.2 Models and Experiments

Baselines include a simple embedding layer trained end-to-end followed by either a linear regression or non-linear MLP head (denoted Simple Embedding + LinReg/MLP), as well as a third baseline that uses pre-trained molecular LMs [10], [11] to embed protein and ligand sequences before concatenation and a linear regression head (denoted LM Embedding + LinReg).

Fine-tuning methods include function composition algorithms of adding a joint non-linear MLP adapter regression head (denoted Joint MLP Head), as well as adding a trainable non-linear adapter to every transformer block before a linear regression head (denoted All layer Adapter Tuning) [30], [31]. Parameter composition algorithms of only updating bias weights (denoted BitFit) in molecular LLMs [32] or only updating weights in a randomly oriented low-dimensional subspace (denoted LoRA) in molecular LLMs [33] before a linear regression head are included. The input composition method of pre-pending trainable embeddings to sequences in molecular LLMs before a linear regression head (denoted Prefix Tuning) is also tested [34].

We also consider a method that combines the Joint MLP Head and BitFit fine-tuning. This is because these two fine-tuning methods showed the most promising results of all fine-tuning methods (as can be seen in Table 4.1) whilst not considerably increasing computation time. This method is denoted ”BitFit & Joint MLP Head”. Key information

Model Name	Performance (RMSE)	Parameters (millions)	Trainable Parameters (%)
Simple Embedding + LinReg	2.022	0.0529M	100%
Simple Embedding + MLP	2.037	0.412M	100%
LM Embedding + LinReg	1.672	10.9M	0.0064%
Joint MLP Head	1.313	11.3M	3.2%
All Layer Adapter Tuning	1.414	12.0M	9.0%
BitFit	1.432	10.9M	0.37%
LoRA	1.482	11.5M	5.3%
Prefix Tuning	1.464	11.0M	0.83%
BitFit & Joint MLP Head	**1.215**	11.3M	3.5%

Table 4.1: Table summarising the performance of various baselines and PEFT methods tested in this paper. Information on the total number of parameters and the number of trainable parameters is also included to consider the computational burden of each model. Embeddings with molecular LMs outperforms embeddings trained end-to-end, and PEFT of LMs improves performance. The Joint MLP Head model is the most effective individual fine-tuning method, whilst BitFit is the most parameter efficient. Combining these methods gives the best performing model BitFit & Joint MLP Head. Note the first three models are baseline approaches.

about the number of trainable parameters and performance of aforementioned models trained on the PDBbind general set and tested on the PDBbind 2016 core set are summarised in Table 4.1.

4.2.3 Pre-trained LLMs vs End-to-end Encoding

By looking at the results of the baseline approaches, it is clear that encoding with pre-trained protein and ligand LMs is far more effective than training a simple embedding layer to encode protein amino acid sequences and ligand SMILES strings before binding affinity prediction. Encoding with molecular LMs followed by a simple linear regression head results in a RMSE of 1.672, compared to 2.022 and 2.037 for both baseline approaches trained end-to-end with a randomly initiated embedding layer, an increase of 0.35 and 0.37 respectively. This suggests that masked language model pre-training on large amounts of unlabelled protein and ligand sequences does capture important

structural and property information that can aid in the prediction of protein-ligand binding affinity.

4.2.4 Benefit of PEFT

We can also see from the results that PEFT of molecular LMs is considerably more effective than just using frozen LMs as encoders. All PEFT methods have an RMSE of 1.482 or less, being at least 0.19 better than the LM Embedding + Linear Regression baseline model. The best individual PEFT method has an RMSE 0.37 lower than the baseline, whilst the combined method (BitFit & Joint MLP Head) is 0.46 lower. This indicates that PEFT of protein and ligand LMs results in more informative protein and ligand sequence embeddings that can be used to produce higher performance predictions of protein-ligand binding affinity.

4.2.5 Best Individual PEFT Approach

Comparing within fine-tuning methods, we can see that the choice of algorithm has a marked impact on the efficacy of deep learning models. Tuning by adding a joint MLP Head on top of the protein and ligand LMs reaches an RMSE of 1.313, 0.99 higher than any other individual PEFT method. As the only model with a non-linear regression head, this shows the importance of jointly learning non-linear relationships between embedded protein and ligand sequences. Between fine-tuning methods that directly modify transformer blocks of LMs instead of the regression head, the differences are a lot smaller but still notable. All layer Adapter tuning is the best method with an RMSE of 1.414, and the worst method is LoRA with a 0.068 higher RMSE at 1.482. Interestingly, there is no clear relationship between the percentage of trainable parameters in methods and performance. For example, the joint MLP Head model and BitFit both outperform LoRA despite having 3.2% and 0.37% of parameters being trainable compared to 5.3% for LoRA. One impact of this is that in situations where computational burden is important, picking a fine-tuning method that adds less trainable parameters may not harm performance.

4.2.6 Performance of Our Combined Model

Overall, the two most promising individual methods are the joint MLP head which has the markedly best performance of individual PEFT methods, as well as BitFit which is

the third best-performing method despite only having 0.37% of all model parameters as trainable. As these two methods are high-performing and work on different parts of the binding-affinity models, we tested combining these PEFT methods in one model- the BitFit & Joint MLP Head model. This model is considerably better than all other methods tested, achieving an RMSE of 1.215, which is 0.98 better than the second best PEFT method we tested. Furthermore, this model has a very reasonable 3.5% of parameters as trainable. This is likely because the joint non-linear MLP regression head can fully take advantage of the more informative protein and ligand embeddings that are produced by keeping all LM bias vectors as trainable in BitFit. Tuning just the regression head and LM transformer blocks in isolation is not sufficient to maximise performance.

4.3 Comparison With Pre-existing Methods

To appreciate how well our models are performing in a wider context, we compare some of the models tested in section 4.1 to some of the most widely used and highest performing models in the literature as described in the literature review section 2.2.

4.3.1 Methods from Literature and Our Models

We include results from six methods in the literature: 1) Autodock Vina which is a physics-based method using interaction features [7]; 2) Random Forest (RF)-score v3 that uses interaction features and the RF algorithm [22]; 3) DeepDTA that uses a convolutional neural network based model on sequence level information [23]; 4) InteractionGraphNet (IGN) that uses a graph neural network approach with higher order structural and interaction features [9]; 5) CAPLA that uses an attention mechanism and binding pocket information [24]; 6) CAPLA-Pred, a version of CAPLA that only uses sequence level information [24]. We chose to compare these methods to the Joint MLP Head model and the BitFit & Joint MLP Head model as described in section 3.3.3. This is because the Joint MLP Head model is the best-performing individual fine-tuning method, and the BitFit & Joint MLP Head model is the best-performing of our tested methods. We also include the LM Embedding + Linear Regression baseline to allow comparison to a method that doesn't fine-tune the pre-trained LMs.

As mentioned in the introduction, some methods use features pertaining to either higher-order structural information of the protein or information on the interaction of

Model Name (Year)	Performance (RMSE)	Higher Order Features (Yes/No)
Autodock Vina (2010)	1.750	Yes
RF-Score v3 (2015)	1.395	Yes
DeepDTA (2018)	1.443	No
IGN (2021)	1.220	Yes
CAPLA (2023)	**1.200**	Yes
CAPLA-Pred (2023)	1.298	No
LM Embedding + LinReg	1.672	No
Joint MLP Head	1.313	No
BitFit & Joint MLP Head	**1.215**	No

Table 4.2: Table summarising the performance of a baseline and the best two models (bottom section of the table) from experiments in section 4.1, as well as notable models from wider literature. The table also includes whether models used higher order (protein structural or protein-ligand interaction) features. The BitFit & Joint MLP Head model is the best peforming model that only considers sequence level information, outperforming all comparable methods by at least 0.083 (7%).

the protein and ligand. This information can aid binding affinity prediction but may not be available in a real world setting [1], so data on whether models take advantage of these features is included in Table 4.2 alongside the RMSE performance of models on the 2016 PDBbind core set.

4.3.2 Comparison of Our Methods to Those in Literature

It can be seen from Table 4.2 that our methods using protein and ligand LLMs compare favourably to notable methods from the literature. Our baseline model of using molecular LMs as encoders followed by a linear regression head outperforms physics-based method Autodock Vina by an RMSE of 0.073, evidencing the utility of using molecular LMs in binding affinity prediction even when they are not fine-tuned. Fine-tuning of molecular LMs however is required to get competitive performance with more recent

methods. Training a Joint MLP Head on LMs gives an RMSE of 1.313, performing better than Autodock Vina, RF-Score v3 and IGN despite only tuning regression heads of LMs and not taking advantage of higher order features. The performance of the BitFit & Joint MLP model that also trains bias vectors in the protein and ligand LMs is particularly impressive. The model is considerably better than any other model that doesn't take advantage of higher-order features, achieving an RMSE of 1.215, which is 0.083 lower than the second best purely sequence-based approach CAPLA-Pred. This makes the **BitFit & Joint MLP Head model state-of-the-art for protein-ligand binding affinity from sequence level information alone**. This result emphasises the potential performance benefits available from taking advantage of pre-training with large volumes of unlabelled data for binding affinity prediction and other interactive molecular property prediction tasks. It also shows that tuning both the protein and ligand LM transformer blocks to create more informative embeddings, and using a joint non-linear regression head to learn complex relationships between embedded sequences, is the best way to maximise prediction power.

4.3.3 Comparison of Our Best Model to Interaction-based Methods

As seen in Table 4.1 consideration of higher-order features generally leads to better performance, with the two best methods from the literature IGN and CAPLA both considering higher-order protein structural or interaction features. CAPLA is the best performing method considered with an RMSE of 1.200, 0.02 lower than IGN. However, CAPLA has an RMSE of only 0.015 lower than the BitFit & Joint MLP Head model despite considering higher-order features, suggesting that using pre-trained protein and ligand LLMs can lead to competitive performance even with models that take advantage of higher-order features. This is potentially because increased knowledge of higher-order molecular structure and properties of sequences is learned from masked language model pre-training. Furthermore, when higher-order features are removed from CAPLA, it's RMSE drops by 0.98 to 1.298, which is considerably lower than the BitFit & Joint MLP model, evidencing the effectiveness of the method.

4.4 Effect of Protein Model Size

This section contains results that investigate the effect of protein LLM size on binding affinity prediction models that use molecular LLM embeddings. We are particularly

interested in whether using larger protein LMs with more parameters results in more information-rich embeddings that can improve the performance of protein-ligand binding affinity prediction.

4.4.1 Different Model Sizes and Experiments

Throughout this paper, the ESM-2 [10] protein LLM is used to embed amino acid sequences. The ESM-2 model comes in a variety of sizes, ranging from a model with 6 layers and 8 million parameters to a model with 48 layers and 15 billion parameters, all trained on unlabelled amino acid sequences with a masked language modelling objective [10]. This paper primarily uses the smallest model with 8 million parameters due to limitations in computational resources, but it is possible that using larger models could result in embeddings that capture more complicated and useful protein features, and therefore enable better protein-ligand binding affinity prediction. To test this we train a series of simple binding affinity prediction models using different sizes of ESM-2 models and test how the protein LM size affects performance. We first test LLM embedding + linear regression baseline models that encode both protein and ligand sequences with molecular LMs before concatenation and a linear regression layer for binding affinity prediction. This directly tests how informative embeddings are as only a linear relationship between features can be used to predict binding affinity. We then repeat this experiment but with a non-linear MLP regression head as described in section 3.3.3, which tests whether embeddings from larger protein LMs are more informative when non-linear feature relations are considered. Results of these experiments are shown in Fig. 4.1. Note that ChemBERTa-2 is used as the ligand LM for all experiments. Unfortunately, we could not repeat these size experiments on the ChemBERTa-2 model as it only has one available size [11].

4.4.2 Effect of Model Size on Performance

As seen in Fig. 4.1 there is a positive relationship between increasing the size of protein LLM and the performance of baseline models with a linear regression head. The model with the 8M parameter protein LM has a RMSE of 1.655, compared to 1.586 and 1.534 for the 35M and 150M protein LM respectively, with decreases of 0.069 and 0.052 as the number of parameters increases. This shows that larger protein LMs capture more informative embeddings that give considerably better binding affinity performance when linear relationships are considered. For models with a non-linear

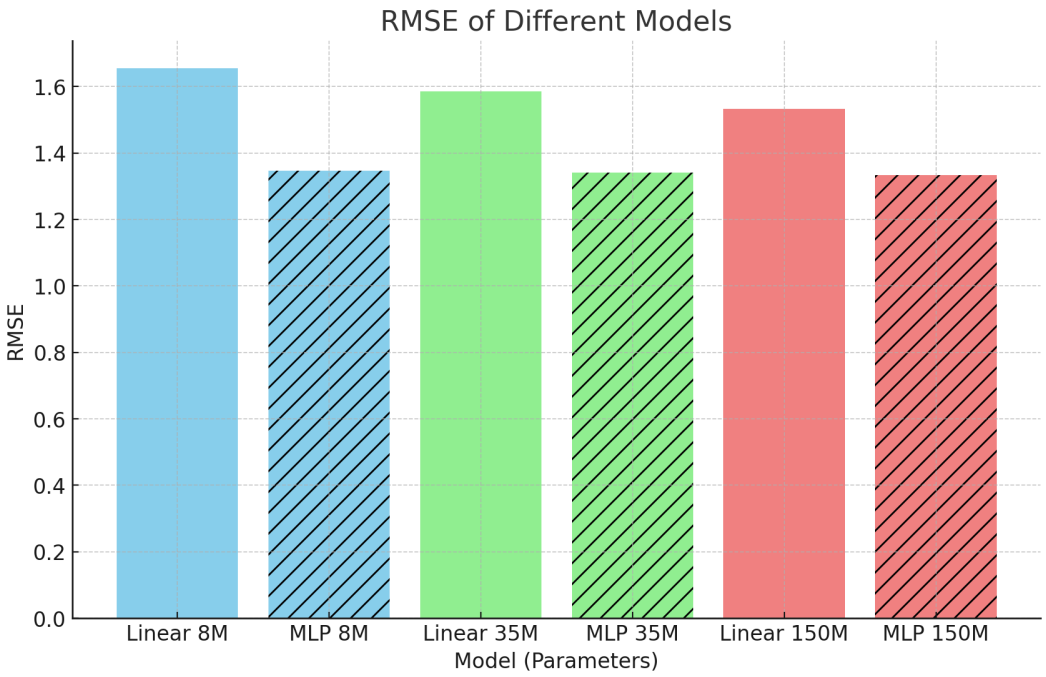

Figure 4.1: Comparison of linear regression head and non-linear MLP regression head applied to embeddings from different sizes of protein LLM. Three sizes of ESM-2 models were used ranging from 8M to 150M parameters (bigger models than this were too computationally expensive to test). Increasing LM size considerably improves performance of models with linear regression heads and only slighly improves performance of models with non-linear MLP heads. Increased LM size gives more informative embeddings, but is less effective and efficient than using PEFT.

MLP head, increasing protein LM model size only very slightly improves performance, with an RMSE of 1.347 using the 8M parameter model, 1.341 with the 35M model and 1.333 with the 150M parameter model. These represent decreases of 0.006 and 0.008 as the number of parameters increases. Increasing protein LM size considerably increases performance with linear regression heads, but not MLP heads, suggesting a more complicated regression head that models non-linear relationships between features can compensate for less linearly informative embeddings. Overall, results show that increasing protein LM size results in more informative embeddings and better binding affinity prediction performance. It is possible that further increasing protein and ligand LM size could lead to even better performance of methods proposed in this paper if computation power allowed. However, increasing protein model size considerably increases model parameters for only moderate performance benefits, these results show PEFT of a smaller protein LM as done in our proposed models is a far more effective

and computationally efficient way to improve performance.

4.5 Relative Importance of Protein and Ligand LLM

We also carry out experiments that determine the general importance of the protein vs the ligand LLM in some of the models we tested in section 4.1 and link performance to LM characteristics. This can help practitioners prioritise where to spend limited computational resources and time to maximise performance gain from using or fine-tuning protein and ligand LMs in binding affinity prediction based on the characteristics of molecular LLMs used.

4.5.1 Using Only One of Protein or Ligand LM

We investigate the relative importance of encoding sequences with protein and ligand LMs in our models by testing methods where only one of the amino acid sequence or ligand SMILES string is encoded by an LLM. In each model, the other sequence will be encoded by a simple embedding layer trained from random initialisation. Binding affinity prediction will be made by concatenating embeddings and using a non-linear MLP regression head as in fine-tuning method 1 (section 3.3.3) so non-linear relations can be learned between embedded features. The model where only the protein sequence is encoded with an LM achieves an RMSE of 1.404, which is 0.039 higher than the model only encoding the ligand sequence with the LM, showing that in our baseline models the ligand LM is more important in improving performance.

4.5.2 Fine-tuning Only One of Protein or Ligand LM

We also test the importance of fine-tuning the protein LLM compared to fine-tuning the ligand LLM when both are present in the binding affinity prediction model. We test the effect of only fine-tuning the protein or ligand LLM on LoRA [33] and prefix tuning [34] PEFT methods as described in section 3.3. In both methods, a linear regression head is used for prediction. When only the protein model is fine-tuned RMSE values of 1.557 and 1.532 are achieved, compared to 1.602 and 1.641 when only the ligand model is fine-tuned, which is a 0.077 average increase between only fine-tuning the protein and ligand LM. So, in our models fine-tuning the protein LM is more important than fine-tuning the ligand LM. Together these results indicate that before fine-tuning the ligand LM ChemBERTa-2 provides more informative features for predicting binding affinity,

likely due to its additional pre-training on ligand property prediction regression tasks [11]. Therefore, there is more performance gain from fine-tuning the protein LM ESM-2 [10] to the task as it has not had additional pre-training to optimise embeddings for property prediction. Results suggest that limited compute budget should be prioritised for fine-tuning LLMs that have not had additional downstream pre-training to optimise embeddings for property prediction tasks. If fine-tuning is not feasible, researchers should try to use molecular LLMs already pre-trained with additional property prediction tasks like in [11] to improve performance.

Chapter 5

Discussion

A consistent theme of the results in this paper is the effectiveness of using protein and ligand large language models (LLMs) and parameter-efficient fine-tuning (PEFT) to address some of the limitations of previous methods and improve the performance of protein-ligand binding affinity prediction.

5.1 Benefits of Using Molecular LLMs and PEFT

Firstly, baselines comparing models with simple embedding layers trained end-to-end to models using protein and ligand LMs to embed sequences were tested. Using protein and ligand LMs pre-trained with masked language modelling on large amounts of unlabelled sequences resulted in far better performance, even with just a simple linear regression head. This indicates that pre-trained protein and ligand LMs do capture important structural and property information of molecules that are present in embedded sequences and this can be leveraged to improve prediction of protein-ligand binding affinity.

We then compared the performance of various PEFT methods applied to protein and ligand LMs to try and improve upon the baseline binding affinity prediction model that used frozen LMs. These included adding a non-linear MLP regression head, adapter tuning [31], BitFit [32], LoRA [33] and prefix tuning [34]. All fine-tuning methods considerably outperformed the frozen LM baseline, indicating that fine-tuning protein and ligand LMs can result in even more informative embeddings with features that are specifically tailored to the task and therefore improve binding affinity prediction.

5.2 Best Methods and State-of-the-art Combined Model

The markedly most effective individual fine-tuning method was adding a joint non-linear MLP regression head to concatenated embeddings of protein and ligand LMs, showing the importance of being able to learn non-linear relations between features present in protein and ligand embeddings to make effective predictions. BitFit, which trains all bias vectors in protein and ligand LMs, was the most efficient method, achieving good performance despite having the lowest percentage of trainable parameters out of all methods at 0.37%. To further increase performance, we created a model that used both a non-linear MLP regression head and BitFit on the protein and ligand LMs, tuning both the transformer blocks in LMs and the regression head. This model was considerably better than all other fine-tuning methods tested, achieving an RMSE of 1.215 on the PDBbind 2016 core set whilst having a reasonably low 3.5% of trainable parameters. This model proves that tuning both LMs to create more relevant and informative embeddings, and non-linear modelling of embedded features is required to achieve the best performance.

The BitFit & Joint MLP Regression Head model also performs very favourably when compared to notable protein-ligand binding affinity prediction models in the literature. Our model outperforms all other methods that don't take advantage of higher-order protein structural or protein-ligand interaction features by a considerable margin, having an RMSE of at least 0.083 (7%) lower than comparable methods. **This makes our model state-of-the-art in predicting protein-ligand binding affinity from sequence-level information alone**. Our model is also competitive with the state-of-the-art model that uses protein-ligand interaction features [24], having an RMSE of only 0.015 higher. These results prove that taking advantage of abundant unlabelled data by using molecular LMs, combined with PEFT to adapt models to a task can overcome limitations created by a lack of high-quality labelled data that limits the performance of most deep learning methods in the literature.

5.3 Effect of Molecular LLM Characteristics

We also carried out some additional experiments into the properties of some of the binding affinity models we created. We found that increasing protein LM size can lead to more informative embeddings which improves the performance of models with both linear and non-linear regression heads. This result suggests that with more

computational power and larger protein and ligand LMs, performance of models in this paper could be improved even further. Performance differences were much more prevalent in models using a linear regression head, suggesting that jointly learning non-linear associations between embeddings can compensate for less linearly informative embeddings produced by smaller protein LMs. Despite increasing protein LM size resulting in improved performance we proved that this is a considerably less effective and computationally efficient way to improve performance than our method of PEFT a smaller model.

We also looked into the relative importance of embedding and fine-tuning with the protein LLM ESM-2 [10] and the ligand LLM ChemBERTa-2 [11] in our models. Only embedding the ligand with ChemBERTa-2 was more effective than only embedding the protein with ESM-2, likely because ChemBERTa-2 additional pre-training on regression tasks resulted in more optimised embeddings for property prediction. As a result of this, there was more performance gain from fine-tuning protein LLM ESM-2 to the binding affinity prediction task than fine-tuning ChemBERTa-2 as ESM-2 embeddings were not optimised for property prediction. These results suggest that when computational resources are limited practitioners should focus on finding LLMs already pre-trained on downstream tasks that require less fine-tuning to achieve good performance. Furthermore, fine-tuning LLMs that have not already been pre-trained on downstream tasks should be prioritised over fine-tuning LLMs that have already had property prediction pre-training.

Chapter 6

Conclusions

In this paper, we test the use of protein and ligand LLMs in predicting protein-ligand binding affinity, with a specific focus on the best PEFT methods and architectures to maximise performance.

Predicting protein-ligand binding affinity is a very important problem due to its potential for real-world application. In particular, screening for strong binders to a protein can result in the discovery of potential new drugs [1], and ML methods for predicting binding affinity have the potential to alleviate the problems of traditional wet-lab experiments for screening, which are costly and time-consuming [5]. Current ML methods used to predict protein-ligand binding affinity have shown potential, achieving good results and outperforming physics-based methods [1]. However, models are inherently limited by a lack of high-quality labelled training data which is expensive to obtain experimentally [5]. To address this issue and improve the performance of binding affinity prediction, we leverage protein and ligand LLMs trained with a masked language modelling objective on large volumes of unlabelled protein amino acid sequences and ligand SMILES sequences [10], [11]. By learning relevant structural and property features of molecules on large amounts of unlabelled data, ML models can take advantage of this pre-training to achieve better performance for binding affinity prediction on smaller amounts of labelled data [12].

To test this hypothesis multiple deep learning methods for binding affinity prediction using protein and ligand LMs were trained on the PDBbind dataset and tested on the PDBbind 2016 core set [14]. Other than baselines, all models used protein LM ESM-2 [10] and ligand LM ChemBERTa-2 [11] to embed sequences before concatenation and a regression head. Various PEFT methods were tested to adapt these models to the task of binding affinity prediction. The most efficient methods were adding a joint non-

linear MLP regression head and BitFit [32], which leaves LM bias vectors as trainable. Combining these two fine-tuning methods produced our best model tested in this paper, achieving an RMSE of 1.215 with a reasonable 3.5% of model parameters being trainable. This result shows fine-tuning both LMs to give more relevant and informative embeddings and fully taking advantage of these by considering non-linear relations of embedding features gives the best binding affinity performance with molecular LMs.

Our Joint MLP Head & BitFit model compares very well to approaches in the literature, considerably outperforming other approaches that don't take advantage of higher-order structural features by at least 7%, making the model the **state-of-the-art purely sequence-based protein-ligand binding affinity prediction model**. This validates our hypothesis that masked language model pre-training on unlabelled data captures features of sequences relevant to binding affinity, and that this reduces the amount of labelled data required to achieve good performance. Results also evidenced that PEFT is an effective method to overcome the computational difficulty of adapting two LLMs to model multiple sequences in interaction properties like binding affinity. Also, results with different sizes of protein LMs proves PEFT is a more effective and efficient way of improving performance than increasing protein LLM size.

The approaches tested in this paper could be easily adapted to improve the performance of other molecule property prediction tasks, especially those involving the interaction of two molecules like in binding affinity prediction. We also carried out further experiments that suggested model performance could be improved even more by considering bigger protein LLMs that capture more informative embeddings, and fine-tuning could be made even more efficient by using molecular LLMs already pre-trained on other downstream property prediction tasks. Future research could explore these LLM characteristics in more detail to make the use of molecular LLMs for binding affinity prediction even more effective and efficient. Future work could also explore how either experimental or predicted higher order structural and interaction features could be incorporated into models that use molecular LLMs for binding affinity prediction, to see if molecular LLMs can also be used to achieve state-of-the-art performance in models using higher order features as well as those only considering sequence level information.

Bibliography

[1] H. Wang, "Prediction of protein–ligand binding affinity via deep learning models," *Briefings in Bioinformatics*, vol. 25, no. 2, p. bbae081, Mar. 2024. [Online]. Available: https://doi.org/10.1093/bib/bbae081

[2] D. Paul, G. Sanap, S. Shenoy, D. Kalyane, K. Kalia, and R. K. Tekade, "Artificial intelligence in drug discovery and development," *Drug Discovery Today*, vol. 26, no. 1, pp. 80–93, Jan. 2021. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7577280/

[3] P. Imming, C. Sinning, and A. Meyer, "Drugs, their targets and the nature and number of drug targets," *Nature Reviews Drug Discovery*, vol. 5, no. 10, pp. 821–834, Oct. 2006, publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/nrd2132

[4] V. S. Kulkarni, V. Alagarsamy, V. R. Solomon, P. A. Jose, and S. Murugesan, "Drug Repurposing: An Effective Tool in Modern Drug Discovery," *Russian Journal of Bioorganic Chemistry*, vol. 49, no. 2, pp. 157–166, 2023. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9945820/

[5] I. Jarmoskaite, I. AlSadhan, P. P. Vaidyanathan, and D. Herschlag, "How to measure and evaluate binding affinities," *eLife*, vol. 9, p. e57264, 2020. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7452723/

[6] D. Sun, W. Gao, H. Hu, and S. Zhou, "Why 90% of clinical drug development fails and how to improve it?" *Acta Pharmaceutica Sinica. B*, vol. 12, no. 7, pp. 3049–3062, Jul. 2022. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9293739/

[7] O. Trott and A. J. Olson, "AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading,"

Journal of Computational Chemistry, vol. 31, no. 2, pp. 455–461, 2010, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jcc.21334. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.21334

[8] P. J. Ballester and J. B. O. Mitchell, "A machine learning approach to predicting protein-ligand binding affinity with applications to molecular docking," *Bioinformatics (Oxford, England)*, vol. 26, no. 9, pp. 1169–1175, May 2010.

[9] D. Jiang, C.-Y. Hsieh, Z. Wu, Y. Kang, J. Wang, E. Wang, B. Liao, C. Shen, L. Xu, J. Wu, D. Cao, and T. Hou, "InteractionGraphNet: A Novel and Efficient Deep Graph Representation Learning Framework for Accurate Protein-Ligand Interaction Predictions," *Journal of Medicinal Chemistry*, vol. 64, no. 24, pp. 18 209–18 232, Dec. 2021.

[10] Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli, A. dos Santos Costa, M. Fazel-Zarandi, T. Sercu, S. Candido, and A. Rives, "Evolutionary-scale prediction of atomic-level protein structure with a language model," *Science*, vol. 379, no. 6637, pp. 1123–1130, Mar. 2023, publisher: American Association for the Advancement of Science. [Online]. Available: https://www.science.org/doi/10.1126/science.ade2574

[11] W. Ahmad, E. Simon, S. Chithrananda, G. Grand, and B. Ramsundar, "ChemBERTa-2: Towards Chemical Foundation Models," Sep. 2022, arXiv:2209.01712 [cs, q-bio]. [Online]. Available: http://arxiv.org/abs/2209.01712

[12] C. Qian, H. Tang, Z. Yang, H. Liang, and Y. Liu, "Can Large Language Models Empower Molecular Property Prediction?" Jul. 2023, arXiv:2307.07443 [cs, q-bio]. [Online]. Available: http://arxiv.org/abs/2307.07443

[13] S. Sledzieski, M. Kshirsagar, M. Baek, R. Dodhia, J. Lavista Ferres, and B. Berger, "Democratizing protein language models with parameter-efficient fine-tuning," *Proceedings of the National Academy of Sciences*, vol. 121, no. 26, p. e2405840121, Jun. 2024, publisher: Proceedings of the National Academy of Sciences. [Online]. Available: https://www.pnas.org/doi/full/10.1073/pnas.2405840121

[14] R. Wang, X. Fang, Y. Lu, C.-Y. Yang, and S. Wang, "The PDBbind Database: Methodologies and Updates," *Journal of Medicinal Chemistry*, vol. 48,

no. 12, Jun. 2005, publisher: American Chemical Society. [Online]. Available: https://doi.org/10.1021/jm048957q

[15] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, "Analyzing Protein Structure and Function," in *Molecular Biology of the Cell. 4th edition*. Garland Science, 2002. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK26820/

[16] S. Kwon and C. Seok, "CSAlign and CSAlign-Dock: Structure alignment of ligands considering full flexibility and application to protein–ligand docking," *Computational and Structural Biotechnology Journal*, vol. 21, pp. 1–10, Jan. 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2001037022005414

[17] P. L. Kastritis and A. M. J. J. Bonvin, "On the binding affinity of macromolecular interactions: daring to ask why proteins interact," *Journal of the Royal Society Interface*, vol. 10, no. 79, p. 20120835, Feb. 2013. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3565702/

[18] Z. Guo and R. Yamaguchi, "Machine learning methods for protein-protein binding affinity prediction in protein design," *Frontiers in Bioinformatics*, vol. 2, p. 1065703, Dec. 2022. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9800603/

[19] V. Govind Kumar, A. Polasa, S. Agrawal, T. K. S. Kumar, and M. Moradi, "Binding affinity estimation from restrained umbrella sampling simulations," *Nature Computational Science*, vol. 3, no. 1, pp. 59–70, Jan. 2023, publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s43588-022-00389-9

[20] K. Yugandhar and M. M. Gromiha, "Protein-protein binding affinity prediction from amino acid sequence," *Bioinformatics (Oxford, England)*, vol. 30, no. 24, pp. 3583–3589, Dec. 2014.

[21] W. A. Abbasi, A. Yaseen, F. U. Hassan, S. Andleeb, and F. U. A. A. Minhas, "ISLAND: in-silico proteins binding affinity prediction using sequence information," *BioData Mining*, vol. 13, p. 20, Nov. 2020. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7688004/

[22] H. Li, K.-S. Leung, M.-H. Wong, and P. J. Ballester, "Improving AutoDock Vina Using Random Forest: The Growing Accuracy of Binding Affinity Prediction by the Effective Exploitation of Larger Data Sets," *Molecular Informatics*, vol. 34, no. 2-3, pp. 115–126, 2015, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/minf.201400132. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/minf.201400132

[23] H. Öztürk, A. Özgür, and E. Ozkirimli, "DeepDTA: deep drug–target binding affinity prediction," *Bioinformatics*, vol. 34, no. 17, pp. i821–i829, Sep. 2018. [Online]. Available: https://doi.org/10.1093/bioinformatics/bty593

[24] Z. Jin, T. Wu, T. Chen, D. Pan, X. Wang, J. Xie, L. Quan, and Q. Lyu, "CAPLA: improved prediction of protein–ligand binding affinity by a deep learning approach based on a cross-attention mechanism," *Bioinformatics*, vol. 39, no. 2, p. btad049, Feb. 2023. [Online]. Available: https://doi.org/10.1093/bioinformatics/btad049

[25] D. S. Fischer, Y. Wu, B. Schubert, and F. J. Theis, "Predicting antigen specificity of single T cells based on TCR CDR3 regions," *Molecular Systems Biology*, vol. 16, no. 8, p. e9416, Aug. 2020. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7418512/

[26] M. Chen, C. J. T. Ju, G. Zhou, X. Chen, T. Zhang, K.-W. Chang, C. Zaniolo, and W. Wang, "Multifaceted protein–protein interaction prediction based on Siamese residual RCNN," *Bioinformatics*, vol. 35, no. 14, pp. i305–i314, Jul. 2019. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6681469/

[27] P.-Y. Libouban, S. Aci-Sèche, J. C. Gómez-Tamayo, G. Tresadern, and P. Bonnet, "The Impact of Data on Structure-Based Binding Affinity Predictions Using Deep Neural Networks," *International Journal of Molecular Sciences*, vol. 24, no. 22, p. 16120, Jan. 2023, number: 22 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/1422-0067/24/22/16120

[28] N. Brandes, D. Ofer, Y. Peleg, N. Rappoport, and M. Linial, "ProteinBERT: a universal deep-learning model of protein sequence and function," *Bioinformatics*, vol. 38, no. 8, pp. 2102–2110, Apr. 2022. [Online]. Available: https://doi.org/10.1093/bioinformatics/btac020

[29] Z. Fu, H. Yang, A. M.-C. So, W. Lam, L. Bing, and N. Collier, "On the Effectiveness of Parameter-Efficient Fine-Tuning," *Proceedings*

of the AAAI Conference on Artificial Intelligence, vol. 37, no. 11, pp. 12 799–12 807, Jun. 2023, number: 11. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/26505

[30] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. D. Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-Efficient Transfer Learning for NLP," in *Proceedings of the 36th International Conference on Machine Learning*. PMLR, May 2019, pp. 2790–2799, iSSN: 2640-3498. [Online]. Available: https://proceedings.mlr.press/v97/houlsby19a.html

[31] Z. Lin, A. Madotto, and P. Fung, "Exploring Versatile Generative Language Model Via Parameter-Efficient Transfer Learning," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, T. Cohn, Y. He, and Y. Liu, Eds. Online: Association for Computational Linguistics, Nov. 2020, pp. 441–459. [Online]. Available: https://aclanthology.org/2020.findings-emnlp.41

[32] E. Ben Zaken, Y. Goldberg, and S. Ravfogel, "BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 1–9. [Online]. Available: https://aclanthology.org/2022.acl-short.1

[33] E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=nZeVKeeFYf9

[34] X. L. Li and P. Liang, "Prefix-Tuning: Optimizing Continuous Prompts for Generation," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds. Online: Association for Computational Linguistics, Aug. 2021, pp. 4582–4597. [Online]. Available: https://aclanthology.org/2021.acl-long.353

[35] J. Vamathevan, D. Clark, P. Czodrowski, I. Dunham, E. Ferran, G. Lee, B. Li, A. Madabhushi, P. Shah, M. Spitzer, and S. Zhao, "Applications of machine learning in drug discovery and development," *Nature Reviews Drug Discovery*,

vol. 18, no. 6, pp. 463–477, Jun. 2019, publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s41573-019-0024-5

[36] A. Dhakal, C. McKay, J. J. Tanner, and J. Cheng, "Artificial intelligence in the prediction of protein–ligand interactions: recent advances and future directions," *Briefings in Bioinformatics*, vol. 23, no. 1, Jan. 2022, publisher: Oxford University Press. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8690157/

[37] "PDBbind+," https://www.pdbbind-plus.org.cn/, accessed: 2024-08-10.

[38] R. Sánchez and A. Sali, "Large-scale protein structure modeling of the Saccharomyces cerevisiae genome," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 95, no. 23, pp. 13 597–13 602, Nov. 1998.

[39] P. J. A. Cock, T. Antao, J. T. Chang, B. A. Chapman, C. J. Cox, A. Dalke, I. Friedberg, T. Hamelryck, F. Kauff, B. Wilczynski, and M. J. L. de Hoon, "Biopython: freely available Python tools for computational molecular biology and bioinformatics," *Bioinformatics*, vol. 25, no. 11, pp. 1422–1423, Jun. 2009. [Online]. Available: https://doi.org/10.1093/bioinformatics/btp163

[40] "Rdkit: Open-source cheminformatics," https://www.rdkit.org, accessed: 2024-08-10.

[41] P. Gaurang, A. Ganatra, Y. Kosta, and D. Panchal, "Behaviour Analysis of Multilayer Perceptronswith Multiple Hidden Neurons and Hidden Layers," *International Journal of Computer Theory and Engineering*, vol. 3, pp. 332–337, Jan. 2011.

[42] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," Jul. 2012, arXiv:1207.0580 [cs]. [Online]. Available: http://arxiv.org/abs/1207.0580

[43] L. Zhuang, L. Wayne, S. Ya, and Z. Jun, "A Robustly Optimized BERT Pre-training Approach with Post-training," in *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, S. Li, M. Sun, Y. Liu, H. Wu, K. Liu, W. Che, S. He, and G. Rao, Eds. Huhhot, China: Chinese Information Processing Society of China, Aug. 2021, pp. 1218–1227. [Online]. Available: https://aclanthology.org/2021.ccl-1.108

[44] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[45] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, u. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[46] J. Howard and S. Ruder, "Universal Language Model Fine-tuning for Text Classification," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, I. Gurevych and Y. Miyao, Eds. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 328–339. [Online]. Available: https://aclanthology.org/P18-1031

[47] W. Qi, Y.-P. Ruan, Y. Zuo, and T. Li, "Parameter-Efficient Tuning on Layer Normalization for Pre-trained Language Models," Dec. 2022, arXiv:2211.08682 [cs]. [Online]. Available: http://arxiv.org/abs/2211.08682

[48] J. Pfeiffer, A. Kamath, A. Rücklé, K. Cho, and I. Gurevych, "AdapterFusion: Non-Destructive Task Composition for Transfer Learning," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, P. Merlo, J. Tiedemann, and R. Tsarfaty, Eds. Online: Association for Computational Linguistics, Apr. 2021, pp. 487–503. [Online]. Available: https://aclanthology.org/2021.eacl-main.39

[49] Y. Zeng and K. Lee, "The Expressive Power of Low-Rank Adaptation." International Conference on Learning Representations, Oct. 2023. [Online]. Available: https://openreview.net/forum?id=likXVjmh3E

[50] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch:

an imperative style, high-performance deep learning library," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., Dec. 2019, no. 721, pp. 8026–8037.

[51] G. Zhang, L. Li, Z. Nado, J. Martens, S. Sachdeva, G. Dahl, C. Shallue, and R. B. Grosse, "Which Algorithmic Choices Matter at Which Batch Sizes? Insights From a Noisy Quadratic Model," in *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/hash/e0eacd983971634327ae1819ea8b6214-Abstract.html

[52] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," Nov. 2017. [Online]. Available: https://www.semanticscholar.org/paper/Decoupled-Weight-Decay-Regularization-Loshchilov-Hutter/d07284a6811f1b2745d91bdb06b040b57f226882

[53] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-demos.6