LONDON'S GLOBAL UNIVERSITY

UCL

Leveraging Deep Learning Techniques for Accurate Prediction of Metal-Binding Sites in Protein Sequences

YQLN5[1]

MEng Computer Science

Supervisor: Prof. David Jones

Submission date: 2nd May 2023

[1]**Disclaimer:** This report is submitted as part requirement for my MEng Degree in Computer Science at UCL. It is substantially the result of my own work except where explicitly indicated in the text. The report may be freely copied and distributed provided the source is explicitly acknowledged.

Abstract

Metal ions play crucial roles in numerous biological processes, and accurately predicting their binding sites in proteins is essential for understanding protein function, folding, and stability. However, this problem is challenging to solve using computational methods due to the comparatively fewer occurrences of metal-binding amino acids than non-binding ones and the diverse patterns of local geometry around these binding sites.

This project leverages the power of protein language models and advanced deep learning architectures to predict the binding sites of various metal ions in protein sequences. The most robust model accurately identifies the binding sites of 11 metal ion classes, achieving competitive or superior performance compared to existing methods, with an average MCC of 0.558, AUROC of 0.96, and AUPRC of 0.539.

In addition to presenting a highly effective model, this dissertation provides valuable insights into methodological choices specifically relevant to metal-binding site prediction studies and offers open-source model development pipelines for easy adoption and extension by other researchers. The research findings underscore the potential of deep learning techniques in addressing biologically relevant problems and contribute useful knowledge to the scientific community, supporting further developments in this domain.

Contents

1 Introduction **1**

1.1 Problem Outline 1

1.2 Motivation 1

1.3 Project Aims & Goals 2

1.3.1 Aims 2

1.3.2 Goals 2

1.4 Project Approach 3

1.5 Report Structure 3

2 Background **4**

2.1 Protein Structure 4

2.1.1 Primary Structure 4

2.1.2 Secondary Structure 5

2.1.3 Tertiary Structure 5

2.2 Metalloprotein 6

2.3 Natural Challenges of Metal-binding Site Prediction 7

3 Literature Review **8**

3.1 Methods 8

3.2 Feature Representations 9

3.3 Tackling Class Imbalance Problem 10

3.4 Evaluation Metrics 11

4 Dataset **12**

4.1 Data Overview 12

4.2 Data Analysis 13

4.3 Data Filtering & Splitting 17

4.3.1 Guidelines 17

4.3.2 Procedures 17

4.4 Dataset Composition 18

5 Methodology **20**

5.1 Label Representation 20

5.1.1 Define the Problem 20

5.1.2 Label Encoding & Extraction . . . 20
5.1.3 An Example . . . 21
5.2 Feature Representation . . . 21
5.2.1 Representation Method Choice . . . 21
5.2.2 Protein Language Model . . . 22
5.2.3 Embedding Generation . . . 22
5.3 Network Design . . . 23
5.3.1 Convolutional Neural Network . . . 23
5.3.2 Transformer Encoder . . . 25
5.4 Model Training . . . 28
5.4.1 Training Settings . . . 28
5.4.2 Length Padding & Masking . . . 28
5.4.3 Loss Function . . . 28
5.4.4 Class Imbalance Cures . . . 29
5.5 Validation . . . 29
5.5.1 Metrics . . . 29
5.5.2 Hyperparameter Tuning . . . 30
5.5.3 Threshold Tuning . . . 31

6 Result Analysis **32**
6.1 Analysis on Model Hyperparameter . . . 32
6.2 Transformer Encoder versus CNN . . . 34
6.2.1 Comparison of Prediction Results . . . 34
6.2.2 Discussion of Learning Curves . . . 35
6.3 Training Longer No Better? . . . 37
6.4 Ankh versus ProtT5 . . . 38
6.5 Tuning the Probability Thresholds . . . 39
6.6 Best Prediction Model from Best Predicted Metal Ions . . . 41
6.7 Comparison with Predictors from Related Work . . . 42

7 Discussion **44**
7.1 Discussion of the Methodologies . . . 44
7.2 Discussion of the Results . . . 45
7.2.1 Low Precision High Recall Issue . . . 45
7.2.2 High Specificity High Accuracy $\neq$ Good Model . . . 45
7.2.3 Why Mn(2+)-Binding Site is Especially Hard . . . 45
7.2.4 Biological Significance of the Model . . . 47

8 Conclusions **48**
8.1 Achievements . . . 48
8.2 Limitations & Future Work . . . 48

A Other Comparison Results **58**

B Development Code and Datasets **61**

Chapter 1

Introduction

1.1 Problem Outline

Metal ions such as zinc, iron, copper, and magnesium play important roles in various biological processes. They can interact with proteins and other biomolecules, affecting their structure, stability, and activity. In particular, metal ions contribute to protein folding and stabilization, as well as the regulation of protein activity and signaling pathways. Furthermore, metal ions can serve as enzyme cofactors, facilitating catalysis and other biochemical reactions [1].

Accurately identifying metal-binding sites in proteins can offer valuable insights into their function and help guide drug design efforts. Numerous computational methods have been developed to achieve this goal, including sequence-based approaches, structure-based approaches, and hybrid techniques that combine both strategies. These methods help pinpoint potential metal-binding sites, which can then be experimentally verified and subjected to further investigation.

1.2 Motivation

The vast majority of known protein sequences in the UniProt [2] database lack metal-binding site annotations, with only 3% of unreviewed proteins in the UniProt/TrEMBL section possessing such information, compared to 18% of curated proteins in the UniProtKB/Swiss-Prot section. This disparity in coverage between reviewed and unreviewed proteins suggests that numerous metal-binding site annotations remain unidentified. In response to UniProt's challenge to detect metal-binding sites [3], this project aims to harness the power of advanced deep learning techniques, including convolutional neural networks (CNN) and transformers, to accurately predict metal-binding sites in protein sequences. The challenge provides a comprehensive dataset[1] that serves as a valuable resource to facilitate this research endeavor.

[1] Available at https://ftp.ebi.ac.uk/pub/contrib/UniProt/prediction_challenges/1_metal_binding/, Creative Commons Attribution 4.0 International (CC BY 4.0) License copyrighted.

1.3 Project Aims & Goals

1.3.1 Aims

The primary objective of this research project is to investigate the current state-of-the-art machine learning methods and apply them to develop models for predicting metal-binding sites from protein sequences. Specifically, the models will be designed to accurately predict the type of metal ions that bind to a given protein sequence and identify the precise locations of the corresponding amino acid residues where the binding occurs.

Moreover, this project aims to evaluate the effectiveness and applicability of the latest machine learning techniques in addressing this challenge and offer valuable insights to other scholars and practitioners in this field.

Ultimately, the outcomes of this study can contribute to the development of more robust and reliable approaches for metal-binding site prediction, which can significantly enhance our understanding of protein structure and function, and advance drug discovery and design.

1.3.2 Goals

In order to achieve the aforementioned aims, the following goals are expected to be accomplished:

- Conduct a comprehensive literature review of the metal-binding site prediction problem, encompassing recent studies and current state-of-the-art (SOTA) approaches.
- Preprocess the datasets, with sufficient size and diversity, that can be used for training and testing the machine learning models. The datasets should be properly split to ensure that the models are tested on unseen data and are not overfitted.
- Implement suitable data representations and network architectures, which should be able to effectively capture the key features of metal-binding sites and produce accurate predictions.
- Train and cross-validate the models built by different variations of the architectures. The models should be optimized for performance metrics, such as precision, recall, and specificity.
- Evaluate and compare the developed models against one another, as well as against SOTAs. A range of evaluation metrics, including but not limited to Matthews Correlation Coefficient (MCC), Area Under the Receiver Operating Characteristic (AUROC), and Area Under the Precision-Recall Curve (AUPRC) should be used to assess the performance of the models.
- Conduct a critical evaluation of the project, including an assessment of its discoveries, achievements, limitations, and potential future directions for improvement.

1.4 Project Approach

The project unfolds in three distinct phases. The first phase involves a comprehensive literature review, where existing research on the subject is thoroughly examined to identify key theories, concepts, and methodologies pertinent to the research question. This foundation informs subsequent stages of the project.

In the second phase, the insights from the literature review and the researcher's expertise are employed to address the specific problem. This phase consists of three substages. Initially, UniProt raw datasets are examined, filtered, combined, and partitioned into non-similar development and independent test sets.

Subsequently, the task is framed as a multi-label classification problem, with annotations encoded into label representations and protein sequences represented using protein language model embeddings.

During the model training stage, the problem is first approached using a naive CNN model, followed by a more advanced transformer encoder designed for this task. The models and their variants are tested and evaluated, with results being academically analyzed and rationalized.

In the project's final phase, the research findings undergo critical review. The approach's limitations are discussed, and suggestions for future research directions are proposed.

1.5 Report Structure

The report is organized into eight chapters, which are outlined as follows:

Chapter One briefly introduces the problem that this project aims to address and the objectives that are sought to be achieved.

Chapter Two provides a background on the chemical and biological concepts that are essential for understanding the problem of metal-binding site prediction.

Chapter Three reviews the related work in this field and emphasizes the main papers that have inspired this project, highlighting their key contributions and limitations.

Chapter Four illustrates the protein sequence data format and describes how the datasets are split and preprocessed to enable their use for machine learning.

Chapter Five explains the methodologies that were used to solve the problem, including feature extraction and representation, and the machine learning techniques that were employed to train and validate the models.

Chapter Six evaluates the results that were obtained from the models and critically analyzes their performance, discussing the strengths and weaknesses of each approach.

Chapter Seven discusses in a more general view the choice of the methodologies and neural network implementation, highlighting their implications for the project outcomes.

Chapter Eight summarizes the key achievements of the project and provides a critical evaluation of its contributions, limitations, and future work that could be pursued.

Chapter 2

Background

2.1 Protein Structure

Proteins are complex macromolecules composed of one or multiple elongated chains of amino acid residues, known as polypeptides. These versatile biomolecules facilitate a plethora of functions within living organisms, including DNA replication, molecular transportation, metabolic reaction catalysis, and cellular structural support [4]. All proteins possess at least primary, secondary, and tertiary structures, with a subset also exhibiting a quaternary structure. The primary structure comprises a linear sequence of amino acids. In the secondary structure, distinct regions of amino acid chains are stabilized by hydrogen bonding within the polypeptide backbone, forming alpha-helices and beta-pleated sheets. The tertiary structure, representing a protein's three-dimensional conformation, is determined by interactions among side chains stemming from the polypeptide backbone. Lastly, the quaternary structure, which affects the overall three-dimensional shape of the protein, arises from interactions between side chains of two or more polypeptide units [5].

2.1.1 Primary Structure

The primary structure of a protein refers to the specific linear sequence of amino acids that constitute the polypeptide chain. Determined by the genetic information encoded in an organism's DNA, the primary structure is essential for establishing the protein's overall shape and function [6]. With 20 different amino acids that can be arranged in numerous combinations, a diverse array of proteins with unique properties and functions can be generated. Fig 2.1 shows the chemical structures and letter encoding of the 20 amino acids.

The primary structure's relationship with metal binding lies in the presence of specific amino acid residues whose side chains can coordinate metal ions. Certain amino acids possess side chains with metal-binding functional groups, such as carboxylic acids, amines, and thiols. When these residues are present and appropriately positioned in the primary structure, they facilitate the formation of metal-binding sites as the protein folds into its higher-order structures [7].

A simple example of a primary structure is the amino acid sequence of a small protein called insulin [9], which is crucial for regulating blood sugar levels in the body. Insulin consists

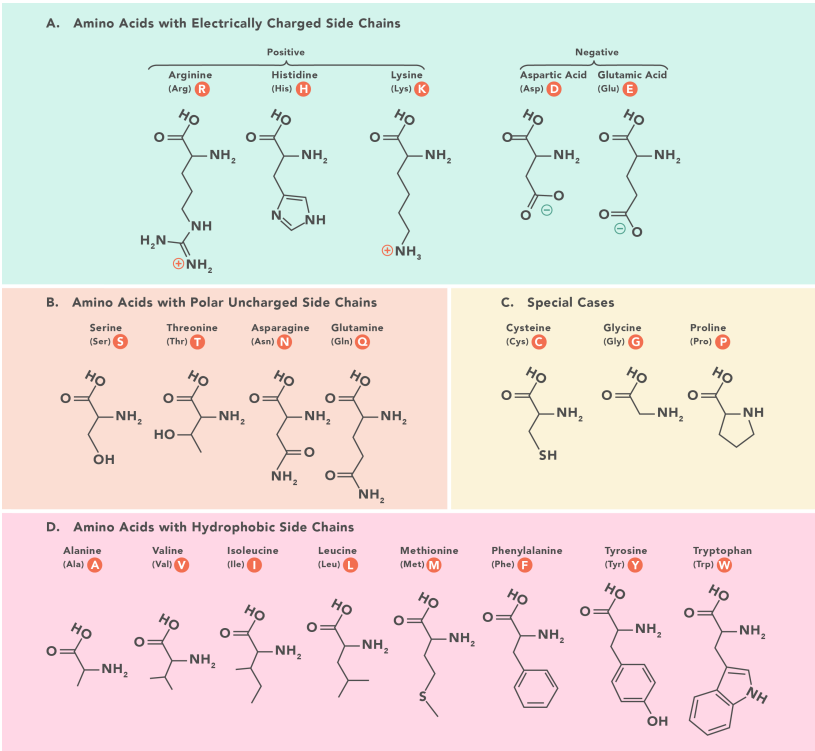


Figure 2.1: The chemical structures of the 20 amino acids that make up proteins and their three-letter and one-letter codes [8].

of two polypeptide chains: chain A, composed of 21 amino acids, and chain B, composed of 30 amino acids. These chains are connected by disulfide bridges, but the primary structure only refers to the amino acid sequence in each chain. The sequence for insulin chain A is:

- Three Letter Code: Gly-Ile-Val-Glu-Gln-Cys-Cys-Thr-Ser-Ile-Cys-Ser-Leu-Tyr-Gln-Leu-Glu-Asn-Tyr-Cys-Asn
- One Letter Code: GIVEQCCTSICSLYQLENYCN

2.1.2 Secondary Structure

Protein secondary structure refers to the local, recurring patterns in a polypeptide chain's three-dimensional conformation, primarily driven by hydrogen bonding between backbone amide and carbonyl groups. Three common secondary structures in proteins are alpha helices, beta sheets composed of beta strands, and turns. Unclassifiable structures typically fall into the "random coil" category [10].

The secondary structure of a protein can play a role in metal binding, as the specific arrangement of amino acid residues can create a suitable environment for metal ions to bind.

2.1.3 Tertiary Structure

The tertiary structure of a protein refers to the overall three-dimensional conformation of the polypeptide chain, determined by the spatial arrangement of secondary structures and stabilized by various types of interactions between the side chains [11].

As a polypeptide chain folds into its tertiary structure, amino acid side chains with metal-binding functional groups can be brought into close proximity, forming metal-binding

sites. The three-dimensional arrangement of these residues creates a specific coordination geometry that accommodates the metal ion, enabling the protein to bind and interact with the metal [12].

2.2 Metalloprotein

Metal-binding proteins, also known as metalloproteins, are proteins capable of interacting with metal ions for various purposes, such as stabilizing their structure, facilitating catalytic activity, or performing other biological functions. The metal ions bound by metalloproteins, referred to as "cofactors", can include transition metals like iron, zinc, copper, manganese, and others [13].

Metal-binding proteins contain specific sites, known as metal-binding sites, where metal ions interact with the protein. These sites are formed by amino acid residues with functional groups that coordinate with metal ions [14], and their arrangement and geometry are determined by the protein's overall structure, including its primary, secondary, and tertiary structures.

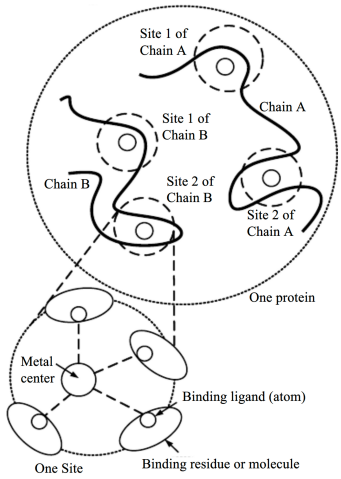


Figure 2.2: The hierarchical structure of a metal-binding protein [15].

Fig 2.2 illustrates the hierarchical model of metalloproteins, consisting of four distinct layers based on physical size: protein, chain, site, and ligand. The protein, at the highest level, is composed of one or multiple polypeptide chains.

Each chain can contain multiple sites, with every site providing coordinate information for the complete metal-binding site. A site comprises molecules, including amino acids and other non-amino acid complexes, surrounding the central metal ion, which constitutes the second coordination shell of the central metal [16]. This project's focus is predicting the locations of these molecules (specifically, amino acid residues) within the protein sequence.

Lastly, any atom directly interacting with the central metal ion is referred to as a "ligand." It is important to note that "ligand" is a versatile term that may refer to different entities depending on the context. In this context, a ligand is an ion or molecule (functional group) that binds to a central metal atom to form a coordination complex [17]. In biochemistry, a

ligand also describes any molecule or atom that binds reversibly to a protein [18]. In this sense, a metal ion is considered a "ligand" binding to the protein chain.

2.3 Natural Challenges of Metal-binding Site Prediction

Metal-binding site prediction is challenging due to the intrinsic nature of the problem, which must be understood before we can develop a model that takes into account these issues accordingly and effectively.

Diversity of metal-binding sites: Metal-binding sites in proteins can vary significantly in terms of coordinating amino acid residues, metal ion types, and overall geometry. This diversity makes it difficult to generalize predictive models or develop a one-size-fits-all approach for identifying metal-binding sites.

Structural data limitations: High-quality structural data of metalloproteins, such as X-ray crystallography or cryo-electron microscopy structures, is essential for developing accurate prediction algorithms. However, the availability of such data is limited, and not all proteins have experimentally determined structures.

Sequence homology limitations: Sequence-based prediction methods typically rely on sequence homology to identify conserved metal-binding residues in homologous proteins. This approach may not always be successful, particularly when the sequence similarity is low or the metal-binding site is not conserved across homologous proteins.

Data imbalance: The number of metal-binding residues is significantly smaller compared to non-binding residues in protein sequences. This imbalance can lead to biased models that perform poorly in identifying metal-binding sites.

Chapter 3

Literature Review

3.1 Methods

Historically, metal-binding site prediction methods have employed various algorithmic approaches, such as geometric [19, 20], energetic [21, 22], and evolutionary-based [23, 24]. Recently, sequence-based metal-binding site prediction has increasingly benefited from machine learning solutions.

AdaBoost (Adaptive Boosting) is an ensemble learning method widely used to address this challenge [25, 26, 27, 28]. Its iterative nature and adaptive weighting of misclassified instances allow it to effectively manage the significant imbalance problem inherent in metal-binding site prediction.

MIonSite [25] introduces an Enhanced AdaBoost (E-AdaBoost) algorithm to further improve its ability to handle imbalanced classes. In each iteration, E-AdaBoost selectively samples a subset from the majority set (non-binding residues) using a sample-distribution-based sampling technique, and combines it with the entire minority set (metal-binding residues) to train base classifiers that produce metal-binding probabilities.

Other classifier ensemble approaches have also proven effective in addressing imbalanced classification issues and have been employed in various metal-binding site prediction studies. For instance, Random Forest (RF), which ensembles a collection of decision trees to vote for binary output, has been used in [29, 30, 28]. Gradient Boosting Machine, which ensembles a weighted sum of weak parameterized functions, has been utilized in [31, 32]. Moreover, Weighted Sparse Representation-based Classifiers have been combined in [33].

The ensemble methods mentioned above typically approach metal-binding site prediction as a binary classification problem. An ensemble classifier scans each residue in a protein sequence, predicting whether it binds to a metal. A common approach involves using a sliding window technique centered on the target residue, with an observation scope determined by the window length. This technique effectively captures local patterns around the sites of interest. However, choosing the appropriate window size is challenging. A small window may not provide sufficient information, while a large window could result in unstable training and unreliable models. Although many studies use validation sets to heuristically determine the optimal window size, the choice often varies among residues. It is important to note that

the sliding window method may miss critical information, as residues distant in the primary structure can be spatially close in the 3D conformation. Another concern is that predicting various metal types requires training multiple independent classifiers, which overlooks potential interrelationships between metal-specific binding predictions.

Compared to traditional machine learning methods, deep learning offers greater scalability and adaptability for larger and more complex datasets. Advances have been made in developing deep learning models to address the metal-binding site prediction problem [34, 29, 24, 35, 25, 15, 27, 36, 37, 38, 39, 40].

mebipred [35] employs a feed-forward Multi-Layer Perceptron (MLP) with back-propagation, representing protein sequences using fixed-length features. However, due to limitations in the expressiveness of the feature representation choice, the model can only effectively classify proteins as metal-binding but cannot identify specific metal-binding sites.

DELIA [36] is based on a hybrid deep neural network that integrates 1D sequence-based Bidirectional Long Short-Term Memory (BiLSTM) with a 2D structure-based CNN. The outputs from both networks are combined via fully connected layers to predict specific binding sites for four metal ions. Despite achieving high precision (above 0.50) for all metal types, the model struggles with data imbalance, resulting in low true positive rates (recall). Notably, DELIA's performance on proteins with predicted structures is substantially diminished, raising concerns about its practical applicability.

bindEmbed21DL [37] utilizes a 2-layer 1D CNN that takes ProtT5 [41] generated protein embeddings as input. This method efficiently predicts whether a residue binds to metal ions, nucleic acids, or small molecules without requiring structural information or multiple sequence alignments (MSAs). However, the naive 1D CNN network design yields less than extraordinary performance, and targeted binding ligands lack specificity.

Ensembles of multiple models have dominated numerous Kaggle[1] competitions. MetalNet [40] employs an ensemble classification model, followed by a graph-based filtering process, to identify metal-binding sites using solely MSA-derived co-evolution information.

3.2 Feature Representations

Feature representation is crucial in protein metal-binding site prediction, as it directly impacts the performance and interpretability of predictive models. Selecting appropriate features can significantly improve prediction accuracy and robustness.

Protein sequence representation methods can be classified into seven categories based on the information they use. Evolutionary information, arguably the most prevalent representation, pertains to conservation patterns and relationships among homologous protein sequences [42]. Derived from MSAs, this information is frequently employed to generate position-specific scoring matrices (PSSMs) or other profiles that capture conservation and substitution patterns [43].

Structural information, including secondary structure [44], relative solvent accessibility (RSA) [45], and dihedral angles [46], reflects local protein structure, providing insights into

[1] Available at https://www.kaggle.com/

stabilization and spatial organization required for metal binding. It is important to note that these features typically depend on prediction tools, the accuracy of which can affect model performance and reliability.

Amino acid physicochemical properties, such as mass-to-charge ratio, hydrophilicity, and hydrophobicity, significantly influence ligand affinity [47, 48]. However, these representations consider amino acids individually, neglecting interrelationships within protein sequences.

Whole-protein features, including amino acid composition and structure-based distance matrices, also serve as representations.

Transformer models [49] have revolutionized natural language processing (NLP), leading to the emergence of auto-regressive protein Language Models (pLMs) that generate biophysically informative embeddings from unlabeled data. Notable pLMs, such as ProtTrans [41] and ESM [50], exhibit strong performance in various downstream tasks [51, 52, 53, 54]. However, the extent to which these embeddings encode useful information remains to be investigated and validated [55].

Table 3.1 summarizes commonly used feature representation methods, as well as those employed in the reviewed works in this section.

Table 3.1: Categorized feature representation methods and their usage in the reviewed works.

Category	Representation	Scope	Used by
Evolutionary information	position specific scoring matrices (PSSM)	per residue	MIonSite[25], DELIA[36], LmetalSite[38], TargetS[26]
	hidden Markov models (HMM) [56]	per residue	DELIA, LmetalSite
	Jensen-Shannon divergence (JSD) [57]	per residue	MIonSite
	5-mer feature	per protein	mebipred[35]
	co-evolution contact map	per residue	MetalNet[40]
Predicted structural information	secondary structure: helix, strand, and coil	per residue	MIonSite, DELIA, TargetS, MLP[15], RF[29]
	relative solvent accessibility (RSA)		MIonSite, DELIA, LmetalSite, mebipred, MLP, RF
	dihedral angle: phi, psi		LmetalSite, RF
Physiochemical property	physical properties: mass, volume, area	per residue	mebipred, MLP
	hydrophilic-hydrophobic property		mebipred, MLP, RF
	polarization charge		mebipred, RF
	chemical classification [58]		MLP
Binding propensity	background binding propensity	per residue	TargetS, MLP
	S-SITE predicted propensity [59]		DELIA
Sequence information	amino acid composition	per protein	mebipred
Tertiary structural information	structure-based distance matrix	per protein	DELIA
PLM embedding	ProtT5	per protein or residue	bindEmbed21[37], LmetalSite, LambdaPP[39]
	ESM-1b		LmetalSite, LambdaPP

3.3 Tackling Class Imbalance Problem

Protein metal-binding site prediction faces significant class imbalance arising from the relatively fewer occurrences of metal-binding amino acids compared to non-binding ones. Traditional training settings often struggle with recall-precision trade-offs as a result. Numerous

efforts have been made to address this issue.

Data Resampling

Resampling techniques are commonly employed to address the class imbalance issue in metal-binding site prediction model training [60].

Specifically, mebipred [35] utilizes random undersampling of negative instances during training and testing phases to effectively balance class distribution, preventing model bias towards the majority class.

In addition to undersampling the majority class, DELIA [36] employs oversampling of positive instances per minibatch. This method replicates positive instances within each minibatch during training, ensuring the model is exposed to a sufficient number of positive instances and reducing the risk of underfitting the minority class.

Ensemble learning is often chosen to tackle this problem. Many studies create multiple base models by combining the positive dataset with different subsets of the negative dataset, ensuring all negative instances are used without information loss.

Weighted Loss Function

Weighted loss functions are especially beneficial in addressing imbalanced class classification problems, as they assign distinct weights to instances from different classes. By imposing higher penalties for misclassifying rare instances, the model learns to focus more on the minority class. Specifically, bindEmbed21 [37] employs a weighted cross-entropy loss function, where individual weights are assigned to each ligand class and optimized to maximize performance in terms of F1 score and MCC.

3.4 Evaluation Metrics

In previous studies, prediction models are typically evaluated using the n-fold cross-validation method. Common evaluation metrics include precision, recall (also known as true positive rate or sensitivity), specificity, and accuracy. Notably, MCC and F-score metrics are favored as they reflect a model's ability to handle class imbalance issues.

In addition to traditional metrics, some measurements have been developed specifically for protein metal-binding prediction tasks. For instance, bindEmbed21 [37] introduces CovOneBind, indicating the fraction of proteins with at least one residue predicted as binding, and its inverse, CovNoBind, representing the fraction of proteins with no binding detected in both predictions and experiments. As suggested by the authors, although these two metrics do not reflect a method's performance for a specific protein, they capture its general applicability. A method that predicts binding residues with lower precision but for more proteins may be preferred over one with higher precision for a limited subset, as the former's predictions can still provide valuable insights.

Chapter 4

Dataset

4.1 Data Overview

This project utilizes the datasets provided by the UniProt Metal Binding Site Challenge [3]. The datasets comprise three FASTA formatted files, one deliberately selected positive set verified by reliable evidence, one large positive set containing all metal-binding proteins in UniProtKB/Swiss-Prot identified by automated prediction tools, and one negative set containing reliable non-binding proteins. Accordingly, two annotation files for the positive sets are provided in TSV format.

The FASTA format is a text-based format for representing the protein sequences using one-letter amino acid codes. In a FASTA file, a sequence commences with a greater-than symbol (“>”), succeeded by a single-line descriptive header of the sequence.

UniProtKB has its own FASTA header format[1], most importantly, the header contains a `UniqueIdentifier` in the second “|” separated section, i.e., the primary accession number of the UniProtKB entry, which is distinct for each protein. The next lines immediately following the header is the protein sequence representation of the amino acid codes of varying lengths. Below shows an example of the protein sequence of accession “Q8I6R7” in UniProtKB.

```
>sp|Q8I6R7|ACN2_ACAGO Acanthoscurrin-2 (Fragment) OS=Acanthoscurria gomesiana OX=115339 GN=acantho2 PE=1 SV=1
DVYKGGGGGRYGGGRYGGGGGYGGGLGGGGLGGGGLGGGKGLGGGGLGGGGLGGGGLGGG
GLGGGKGLGGGGLGGGGLGGGGLGGGGLGGGKGLGGGGLGGGGLGGGRGGYGGGGYGGGY
GGGYGGGKYKG
```

In each annotation file is included all the metal-binding information of the positive protein instances in the corresponding FASTA file. Each piece of annotation is in the format:

`Accession<TAB>Evidence<TAB>ChEBI-ID<TAB>Position`

where `Accession` corresponds to the Uniprot’s protein `UniqueIdentifier`, `Evidence` describes the type of evidence and assertion method of the annotation [61], `ChEBI-ID` refers to the ChEBI identifier [62] of the metal ion that binds to the protein, and `Position` labels the position of the amino acid where the metal ion binds in the protein, starting at 1.

By using the Python packages `biopython` and `pandas`, the protein accession and sequence representation, and its corresponding binding metal types and binding positions can be

[1] Available at https://www.uniprot.org/help/fasta-headers

extracted and unified.

To ensure the reliability of the model, this project decides to use only the portion of positive protein instances that are curator annotated. In the rest of the report, the data studied and utilized refer to the positive and negative protein instances that are well annotated only, any automated method annotated data are temporally discarded and may be used in future work.

4.2 Data Analysis

Data composition

The datasets used by this project contain 195450 proteins in total. 2317 are labeled with at least one metal-binding site, 193133 are unlabeled, meaning they are non-binding proteins.

Metal classes

In UniProtKB, proteins can bind to 29 different metal ion classes (Tab 4.1), plus non-binding making up 30 classes in total. It should be noticed that some classes are parent classes with others being the subclasses of them. This information is indicated by the `ChEBI-ID parents` column. Fig 4.1 shows the Venn diagram illustrating the hierarchical relationships of the classes.

It can be observed that all metal ions are positively charged. Iron-sulfur clusters are typically ligated by protein residues coordinated to each of the Fe cation sites [63]. This provides insights that amino acids that are negatively charged are more likely to be the binding sites, e.g., R, K, H.

Metal-binding residues

Fig 4.2 shows the numbers of metal-binding residues for each metal ion class in the positive dataset used in this project. As can be observed in the figure, the most common metal-binding residues are the Zn(2+) binding ones, i.e., 5788 residues, which makes up 30.8% of total metal-binding residues. Hg-binding residues are the least, i.e., only 2 of them. The number of non-binding residues is still dominant over all metal-binding residues, making up more than 99% of total residues of all proteins. This agrees with the statement that metal-binding site prediction is a severe class imbalance problem.

Noted that not all metal ion classes in UniProtKB show up in the positive dataset. Out of 29 classes in total, only 23 of them present in the dataset this project is using.

Class coalescing

Some specific metal-binding samples are too few for machine learning, so we merge the rare or unusual metal ion classes, i.e., [3Fe-4S] cluster, [Ni-4Fe-4S] cluster, [Ni-4Fe-5S] cluster, Cd(2+), Hg(2+), and Co(3+), to their parent classes, i.e., iron-sulfur clusters, a divalent metal cation, and a metal cation according to their characteristics. After coalescing there are now 18 metal ion classes plus 1 non-binding class.

Table 4.1: Metal classes presented in the UniProt datasets.

#	ChEBI-ID	Name	ChEBI-ID Parents
0	CHEBI:48775	Cd(2+)	CHEBI:60240,CHEBI:25213
1	CHEBI:29108	Ca(2+)	CHEBI:60240,CHEBI:25213
2	CHEBI:48828	Co(2+)	CHEBI:60240,CHEBI:25213
3	CHEBI:49415	Co(3+)	CHEBI:25213
4	CHEBI:23378	Cu cation	CHEBI:25213
5	CHEBI:49552	Cu(+)	CHEBI:23378,CHEBI:25213
6	CHEBI:29036	Cu(2+)	CHEBI:23378,CHEBI:60240,CHEBI:25213
7	CHEBI:60240	a divalent metal cation	CHEBI:25213
8	CHEBI:190135	[2Fe-2S] cluster	CHEBI:30408
9	CHEBI:24875	Fe cation	CHEBI:25213
10	CHEBI:29033	Fe(2+)	CHEBI:24875,CHEBI:60240,CHEBI:25213
11	CHEBI:29034	Fe(3+)	CHEBI:24875,CHEBI:25213
12	CHEBI:30408	iron-sulfur cluster	NaN
13	CHEBI:49713	Li(+)	CHEBI:25213
14	CHEBI:18420	Mg(2+)	CHEBI:60240,CHEBI:25213
15	CHEBI:29035	Mn(2+)	CHEBI:60240,CHEBI:25213
16	CHEBI:16793	Hg(2+)	CHEBI:60240,CHEBI:25213
17	CHEBI:49786	Ni(2+)	CHEBI:60240,CHEBI:25213
18	CHEBI:60400	[Ni-Fe-S] cluster	CHEBI:30408
19	CHEBI:47739	[Ni-4Fe-4S] cluster	CHEBI:60400,CHEBI:30408
20	CHEBI:29103	K(+)	CHEBI:25213
21	CHEBI:29101	Na(+)	CHEBI:25213
22	CHEBI:49883	[4Fe-4S] cluster	CHEBI:30408
23	CHEBI:21137	[3Fe-4S] cluster	CHEBI:30408
24	CHEBI:29105	Zn(2+)	CHEBI:60240,CHEBI:25213
25	CHEBI:177874	[Ni-4Fe-5S] cluster	CHEBI:60400,CHEBI:30408
26	CHEBI:21143	[8Fe-7S] cluster	CHEBI:30408
27	CHEBI:60504	[8Fe-9S-C-homocitryl] cluster	CHEBI:30408
28	CHEBI:25213	a metal cation	NaN

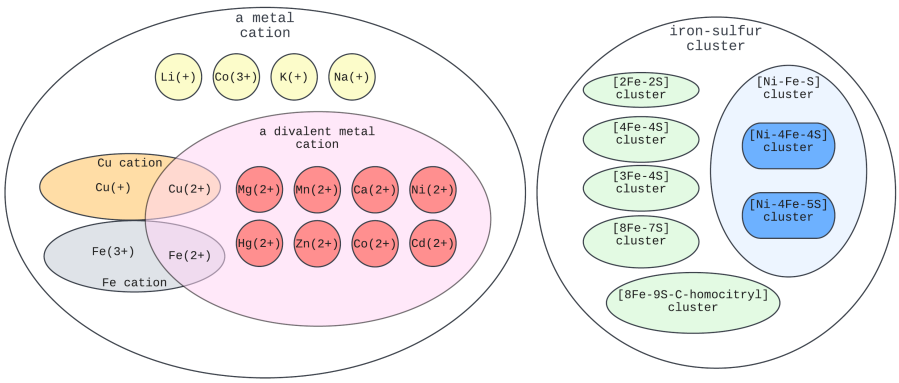


Figure 4.1: A Venn diagram showing relationships of the metal ion classes.

Metal-binding proteins

After examining the extracted protein sequences and their annotations from the raw datasets, some discoveries are made.

Fig 4.3 shows the distribution of the number of metal-binding residues per metalloprotein, highlighting the minimum, maximum, mean, and standard deviation. Metalloproteins in the dataset usually have a small number of metal-binding sites, i.e., 7.79 binding residues on average. Notably, protein accession O31526 has 88 metal-binding residues on it, which is the most among all.

The metalloproteins normally bind to only one or two types of metal ions, and rarely bind to three types and no more than that.

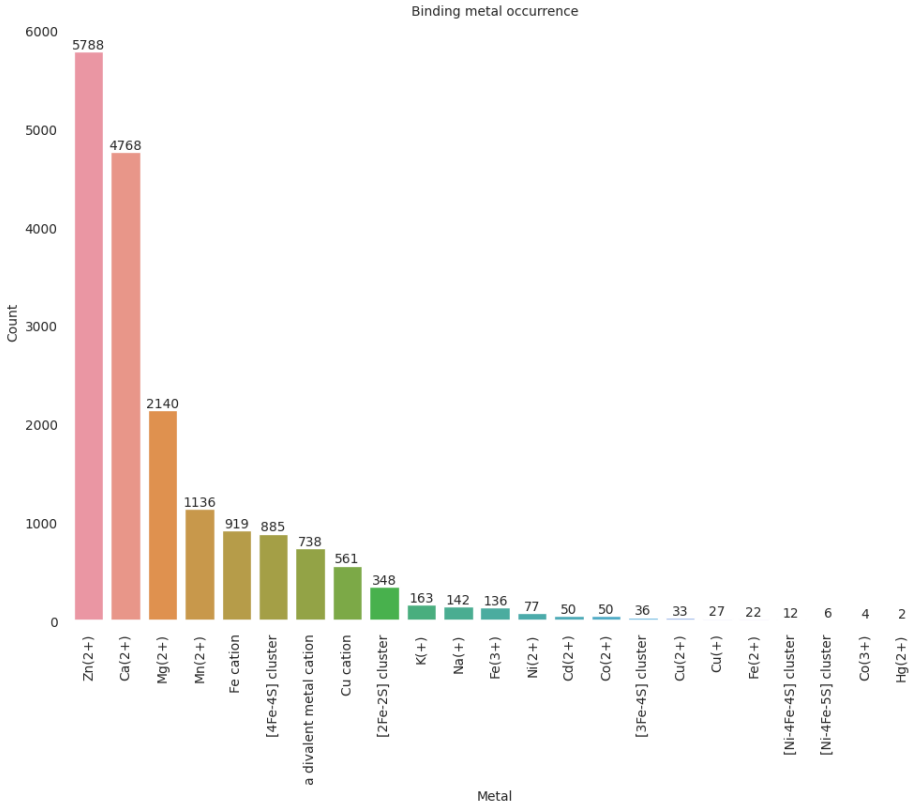


Figure 4.2: The occurrence of metal-binding residues.

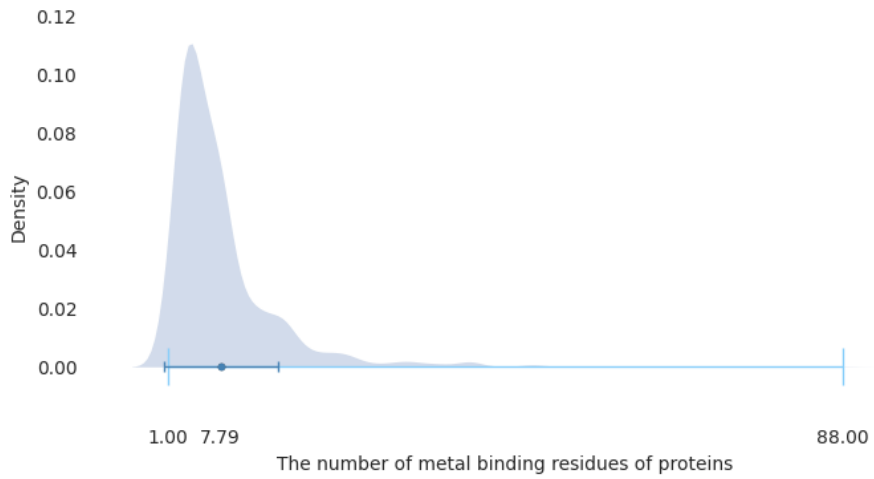


Figure 4.3: Distribution of the number of metal-binding residues per protein visualized, light blue bar shows the range and dark blue bar shows the mean and standard deviation of the number of metal-binding residues of a metalloprotein.

It is surprising that a single residue of a metalloprotein can bind to different types of metal ions, e.g., protein accession "A0A023GPI8" binds to both Mn(2+) and Ca(2+) at position 10. The reasons may be that either the residue is bound by the two metal ions simultaneously via different ligands, or by one or the other under varying chemical environments.

Protein length analysis

The proteins in the dataset vary in length, i.e., the number of composite amino acids. As shown in Fig 4.4, the longest protein is of length 18141, while the shortest is merely 2, with an interquartile range (IQR) of 271. The average length is 430 with a standard deviation of 383. Data of various sizes impose difficulty on training neural networks. Sliding window technique, segmenting sequence, or padding per batch should be conducted to cope with the issue.

The correlation of protein length and the number of metal-binding residues are examined. It can be observed from Fig 4.5, there is no significant sign showing that the protein length and the number of metal-binding residues are statistically related.

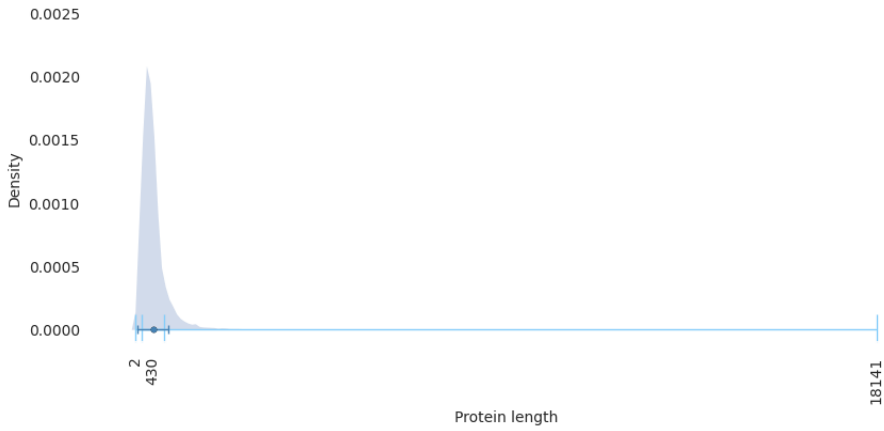


Figure 4.4: Distribution of the protein lengths in the datasets visualized, light blue bar shows the range and IQR, dark blue bar shows the mean and standard deviation of the protein lengths.

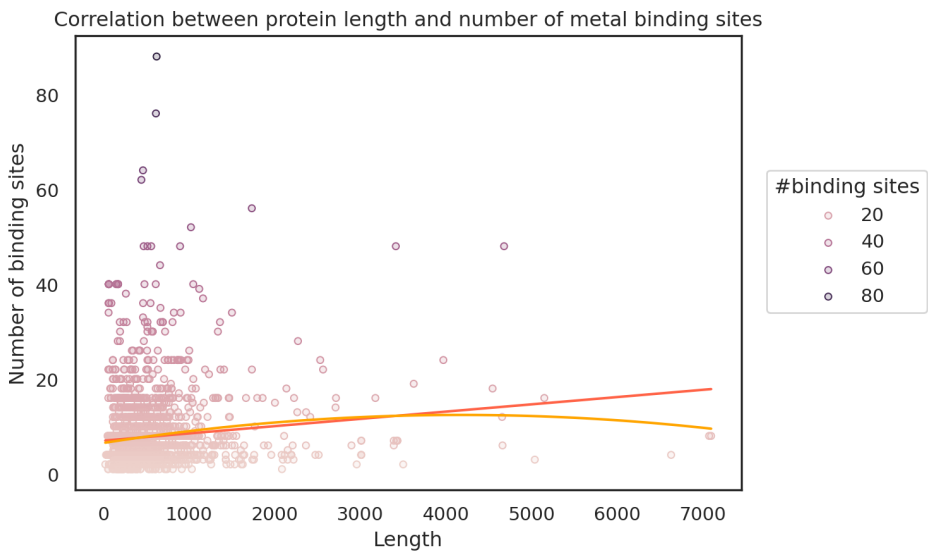


Figure 4.5: Correlation between protein length and the number of binding residues. Red curve shows a fitted first-order polynomial, orange curve shows a fitted second-order polynomial.

4.3 Data Filtering & Splitting

4.3.1 Guidelines

The analysis discloses several characteristics of the metal-binding protein data. First of all, the number of negative protein instances overwhelmingly outweighs the number of positive protein instances, while the latter is what we are more interested in. Therefore, a filter of homologous negative samples is necessary to reduce the redundancy and prevent potential overfitting towards the majority of non-binding protein patterns.

In light of the considerable number of metal ion classes, a proper data splitting strategy is demanded to ensure binding annotations for every metal ion type should show up in both train and test sets. This will guarantee that the test set can provide a fair evaluation of model performance.

Lastly, homologous proteins, i.e., proteins with a common evolutionary origin, usually share similar sequence representation patterns. To better evaluate the robustness and generalizability of a prediction model, the train and test sets should be carefully split to restrict the presence of highly homologous proteins between the train and test sets.

4.3.2 Procedures

The UniProt raw datasets are combined and split into two: one development set (or train set) and one independent test set with a ratio of roughly 9:1. Proteins in the independent test set are of below 40% pairwise sequence identity to those in the development set to make sure the evaluation results will not be biased on the test set. The splitting procedure is as follows:

1. Cluster all the proteins using mmseqs [64] via the command

   ```
   mmseqs easy-cluster combined.fasta clustered tmp --min-seq-id 0.4 --cov-mode 5 -c 0.25  -s 7
   ```

   where `combined.fasta` is the FASTA file containing both positive (metal-binding) and negative (non-binding) proteins.
   `--min-seq-id 0.4` specifies clustering sequences at 40% sequence identity.
   `--cov-mode 5 -c 0.25` sets the alignment mode to allow for short sequences to be at least 25% of the other sequence length. By setting the coverage rate as low as 0.25, even a sequence much shorter than others in a cluster can be counted into that cluster. This choice is made because the clustering is not for the purpose of determining orthology but for machine learning. Machine learning methods, such as CNN, excel in local feature recognition. Clustering following this standard ensures that if a short sequence resembles a segment of some other protein, they are still considered homologous.
   `-s 7` sets the sensitivity of the prefiltering to be 7, i.e., highly sensitive, so that sequence pairs with a low sequence identity threshold can be identified.

2. Negative data, i.e., non-binding proteins, are filtered by retaining only the representatives of clusters that exclusively contain negative instances. This filtering process does not remove clusters with positive instances, as positive data are scarce and each

instance is valuable. Additionally, negative instances similar to positive metal-binding proteins are retained, as they are considered challenging to distinguish and crucial for the model's learning process.

3. Clusters are randomly assigned to development and test sets while ensuring that as many types of metal ions as possible are presented in the test set. The ratio of binding residues for each metal ion class in the test set is maintained at approximately 1:9 to those in the development set. This allocation is performed using a greedy approach.

It is worth noting that many previous studies perform redundancy reduction by limiting pairwise sequence identity to below 40% or even 30% on their entire dataset, containing both negative and positive instances, without providing justification. We choose not to follow this approach because, in the context of metal-binding site prediction, it cannot be guaranteed that two homologous proteins will bind to the same metal ions or share identical binding sites. Aggressively filtering the entire dataset may result in the loss of important information.

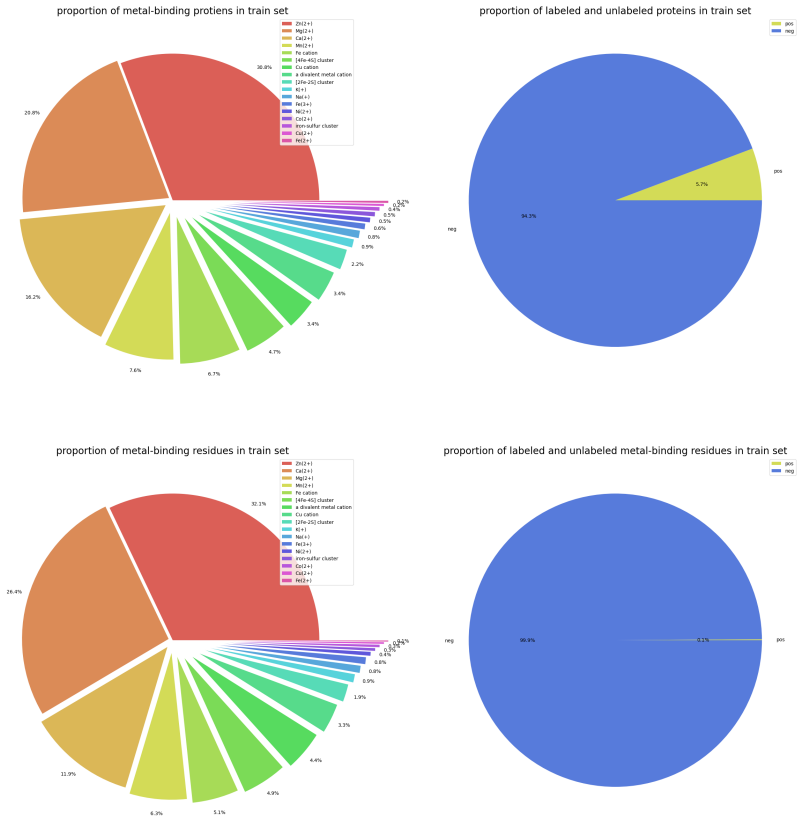
4.4 Dataset Composition

Tab 4.2 summarizes the compositions of the train and test sets used for model training and testing in this project, while Fig 4.6a and Fig 4.6b further visualize the metal-binding protein and residue proportions of both sets.

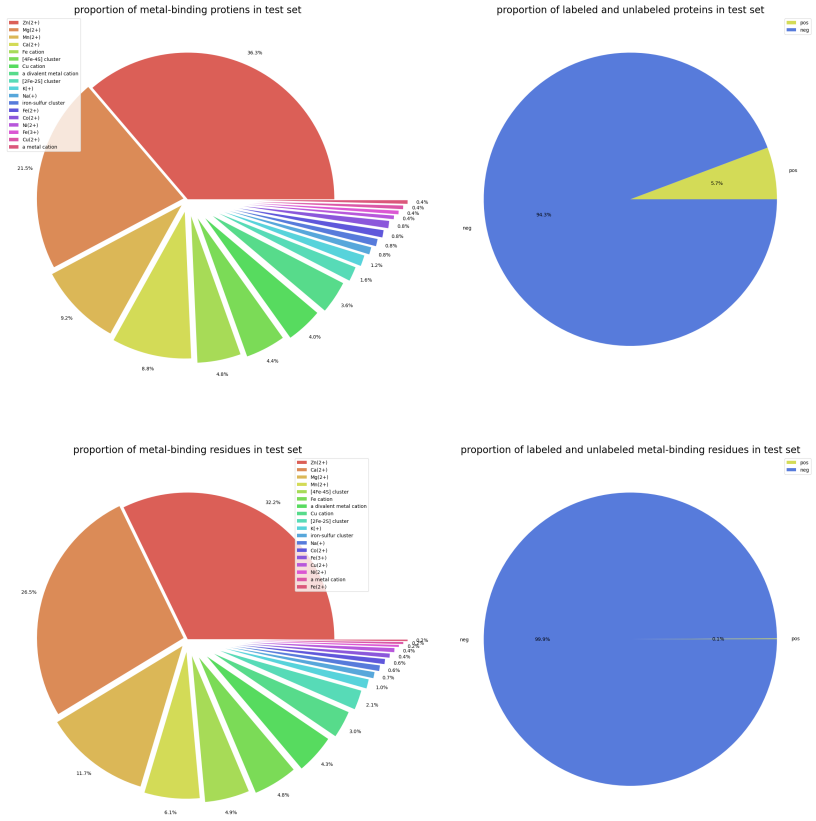
The compositions are calculated without considering the class hierarchy. One may notice an unusual zero annotation for a metal cation in the train set. However, this does not imply that no protein in the train set is bound by a metal cation, given the presence of annotations of its subclasses.

Table 4.2: Train and test set data compositions

		Training dataset		Independent test dataset	
#metal-binding protein		2093		224	
#non-binding protein		34364		3687	
#pos/#neg		0.0609		0.0608	
CHEBI-id	Metal name	#protein	# residue	#protein	# residue
CHEBI:29105	Zn(2+)	694	5215	91	573
CHEBI:18420	Mg(2+)	476	1932	54	208
CHEBI:49883	[4Fe-4S] cluster	110	798	11	87
CHEBI:29108	Ca(2+)	391	4297	22	471
CHEBI:29035	Mn(2+)	172	1028	23	108
CHEBI:60240	a divalent metal cation	77	713	9	77
CHEBI:24875	Fe cation	158	833	12	86
CHEBI:190135	[2Fe-2S] cluster	53	310	4	38
CHEBI:23378	Cu cation	77	535	10	53
CHEBI:29103	K(+)	20	145	3	18
CHEBI:49786	Ni(2+)	12	73	1	4
CHEBI:29101	Na(+)	19	131	2	11
CHEBI:29034	Fe(3+)	15	128	1	8
CHEBI:30408	iron-sulfur cluster	8	42	2	12
CHEBI:29036	Cu(2+)	5	25	1	8
CHEBI:29033	Fe(2+)	4	19	2	3
CHEBI:48828	Co(2+)	10	40	2	10
CHEBI:25213	a metal cation	0	0	1	4



(a) Train set composition



(b) Test set composition

Figure 4.6: Metal-binding protein and metal-binding residue proportions visualized

Chapter 5

Methodology

5.1 Label Representation

5.1.1 Define the Problem

Most previous works treat metal-binding site prediction as multiple binary classification problems, i.e., one binary predictor per metal ion class. However, inspired by the annotated metal hierarchy from UniProt, this project believes it is more natural to treat it as a united multilabel classification problem. There are two advantages of doing so, one is that the model can learn any inter-relationships among the classes in the neural networks, and the other is that it makes the model more robust, e.g., even though a model fails to identify a specific binding metal, it is given a double chance of predicting a more general "a metal cation".

5.1.2 Label Encoding & Extraction

As a multilabel classification problem, metal-binding site prediction has 18 labels, each representing one of the metal classes as discussed in Chapter 4. To convert the raw annotations into labels in matrix format so that machine can understand, the metal classes are encoded as integers from 0 to 17. The encodings are shown in Tab 5.1.
The labels are extracted from the annotation file in the following procedures:

Table 5.1: 18 metal ion classes and their label encodings.

Metal class	Zn(2+)	Mg(2+)	[4Fe-4S] cluster	Ca(2+)	Mn(2+)	a divalent metal cation
Label encoding	0	1	2	3	4	5
Metal class	Fe cation	[2Fe-2S] cluster	Cu cation	K(+)	Ni(2+)	Na(+)
Label encoding	6	7	8	9	10	11
Metal class	Fe(3+)	iron-sulfur cluster	Cu(2+)	Fe(2+)	Co(2+)	a metal cation
Label encoding	12	13	14	15	16	17

1. Given a protein of length L, establish a zero-filled matrix of shape $(L, 18)$, whose first dimension represents the amino acids of the protein, and second dimension represents the metal ion classes.

2. Retrieve the annotations of the protein by its `Accession` from the annotation file.

3. For each piece of annotation, extract the binding metal's `ChEBI-ID` and `Position`.

4. Locate the columns corresponding to the metal ion class and all its parent classes, label the row at the index `Position`−1 by 1 for each column of interest.

5.1.3 An Example

The protein of accession "P08200" is a metal-binding protein of length 416. In the annotation file, its annotations are presented as

```
P08200<TAB>ECO:0000269<TAB>CHEBI:18420<TAB>307
```

which specifies that the protein is bound by CHEBI:18420, i.e., Mg(2+), at position 307.

According to the class hierarchy, it can be deduced that the 307th residue can also be said to be bound by CHEBI:60240, i.e., a divalent metal cation and CHEBI:25213, i.e., a metal cation.

The label matrix is constructed accordingly such that at row index of the position of the metal-binding residue minus one, i.e., 306, and column indices of the encoding of Mg(2+), a divalent metal cation, and a metal cation, i.e., 1, 5, and 17, the elements are set to 1, and all others 0, as visualized in Tab 5.2.

Table 5.2: The label matrix of protein "P08200"

	Zn(2+)	Mg(2+)	[4Fe-4S] cluster	Ca(2+)	Mn(2+)	a divalent metal cation	Fe cation	[2Fe-2S] cluster	Cu cation	K(+)	Ni(2+)	Na(+)	Fe(3+)	iron-sulfur cluster	Cu(2+)	Fe(2+)	Co(2+)	a metal cation
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
306	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1
307	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
415	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5.2 Feature Representation

5.2.1 Representation Method Choice

Features are obtained from raw protein sequences by processing them through a pre-trained protein language model. These models, based on deep learning, employ natural language processing techniques, specifically attention mechanisms and transformers, and are trained on self-supervised tasks over huge corpora of unlabeled protein sequences [65]. The embeddings derived from the final layers of the protein language model have been proven to effectively capture constraints pertinent to protein structure and function. It is worth experimenting

with their performance against this specific task, serving as an evaluation of their effectiveness in transfer learning in the meantime.

Compared to traditional feature representations discussed in Section 3.2, protein language model (pLM) embeddings offer several advantages.

Firstly, traditional features like PSSM or secondary structure capture only a single aspect of protein properties, such as functional or structural information, and may not fully encompass the richness of this information. In contrast, pLMs can capture complex patterns and long-range dependencies in protein sequences, providing a more comprehensive representation of protein structure and function.

Secondly, many traditional feature extraction methods are highly sensitive to protein homology, potentially limiting their generalization to unseen or rare protein sequences. On the other hand, pLMs demonstrate stronger generative capabilities [66]. Finally, although pLM feature extraction requires significant computational resources, it is efficient and time-saving in terms of installation procedures and processing speed.

5.2.2 Protein Language Model

Ankh

Ankh [67] is a state-of-the-art protein Language Model (pLM) published in January 2023. Based on the encoder-decoder transformer architecture, it is trained on the UniRef50 dataset with an embedding dimension of 1536 [68]. Notably, the model prioritizes accessibility by employing a reduced number of training parameters. To counterbalance any potential performance degradation due to this design choice, knowledge-guided experimentation is conducted to optimize the model design.

ProtT5-XL-U50

ProtT5-XL-U50 [41], a.k.a., ProtT5, an encoder-decoder model from the ProtTrans T5 family, has an embedding dimension of 1024 and has been widely utilized in computational biology domains. This model has demonstrated its ability to provide highly informative embeddings for various downstream tasks, such as secondary structure prediction, outperforming other ProtTrans models. Notably, multiple studies on metal-binding site prediction [37, 39, 38] have selected this model for feature extraction.

In this project, the Ankh model serves as the primary feature extraction tool, while ProtT5-XL-U50 is employed as a comparative reference.

5.2.3 Embedding Generation

The maximum input length for protein sequences is restricted to 512 residues. Sequences exceeding this length are segmented into smaller fragments before being input into the protein language model (pLM). This constraint is implemented for two reasons: to accommodate network training requirements for fixed input sizes and to reduce computational complexity, as the longest sequence in the development set surpasses 10,000 residues (as discussed in Section 4.2).

The protein truncation decision stems from two factors. First, the pLMs used in this project have maximum input lengths during training, specifically 512 for Ankh and 1024 for ProtT5. Although these models can theoretically handle proteins of arbitrary lengths, the quality of embeddings for proteins significantly exceeding the training input size limit (e.g., over 10,000) has not been thoroughly examined in the literature. Second, metal-binding sites are heavily influenced by local rather than global protein structures [69], so retaining entire long sequences might introduce excessive information for prediction.

However, it is essential to note that protein folding can cause amino acids distant from one another in the sequence to become proximal in the 3D structure. To address this concern, the maximum length is set to 512 residues, which is 82 amino acids longer than the average protein length in the dataset, mitigating this issue.

The truncated raw protein sequences are input into the pLM, and the output embeddings from the last hidden layer are extracted as features for protein metal-binding site prediction. For a protein of length L composed of L amino acids, the output embedding has dimensions of $(L, 1536)$ for Ankh and $(L, 1024)$ for ProtT5. Fig 5.1 illustrates the embedding generation workflow using the Ankh model.

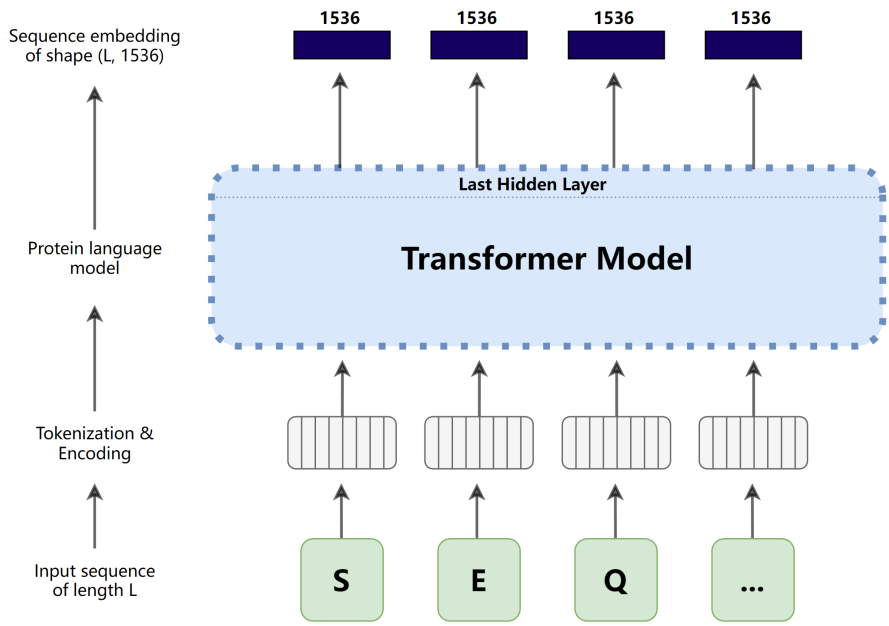


Figure 5.1: Workflow of Ankh embedding generation: a raw sequence is tokenized and encoded, fed into the transformer model, where the outputs of the last hidden layer are extracted as the sequence embedding.

5.3 Network Design

5.3.1 Convolutional Neural Network

CNNs are adept at capturing local patterns and contextual information in the input data. metal-binding sites are often determined by the local environment and interactions of amino acids, which makes it obvious that CNN is a suitable choice for this specific task.

Network architecture

A two-layer 1D CNN architecture is employed as visualized in Fig 5.2. The first layer accepts input protein sequence embeddings of dimensions (L, D). The sequence length L is restricted to 512, and sequences shorter than this length are zero-padded at the end. Each element within the sequence corresponds to a feature vector with dimension D.

The kernel possesses two dimensions: kernel size (K) and feature dimension (D). The kernel size represents the filter's length along the sequence, while the feature dimension is aligned with the input's feature dimension. The filter slides across the sequence, processing a window of K consecutive elements simultaneously.

Kernel size is a crucial hyperparameter as it defines the receptive field surrounding an amino acid. Intuitively, a larger kernel size allows the network to consider more neighboring amino acids when making predictions.

For metal-binding predictions, the kernel must examine each amino acid. To facilitate this, the stride is set to 1.

To maintain length invariance throughout the network, the padding size is always $(K - 1)/2$ for an odd kernel size K.

Multiple filters are incorporated in the 1D CNN layer, enabling each filter to capture distinct patterns in the input sequence.

The output of the layer is a 2D tensor with dimensions $(L,\ C)$, where C denotes the number of filters, or hidden channels. Each element in the output sequence symbolizes the result of the convolution operation for a particular filter and input position.

Before being fed into the second 1D CNN layer, the outputs undergo activation by an Exponential Linear Unit (ELU). The second 1D CNN layer accepts input of size (L, C) and generates an output of $(L, 18)$, corresponding to the number of metal ion classes it aims to predict.

Hyperparameters

The CNN network design introduces two important hyperparameters that are specific to the network:

- Kernel size: The kernel size determines the scope of the local environment that the model uses to decide whether an amino acid residue is metal-binding.
- Number of filters/hidden channels: The number of filters or hidden channels is a hyperparameter that determines how many distinct local patterns the model should learn about one amino acid. The more filters the model has, the more complex patterns it can learn.

The kernel size and the number of filters can affect CNN's ability to extract meaningful features from the input data and its generalization ability. Therefore, careful selection and tuning of these hyperparameters are crucial for optimizing the performance of the CNN.

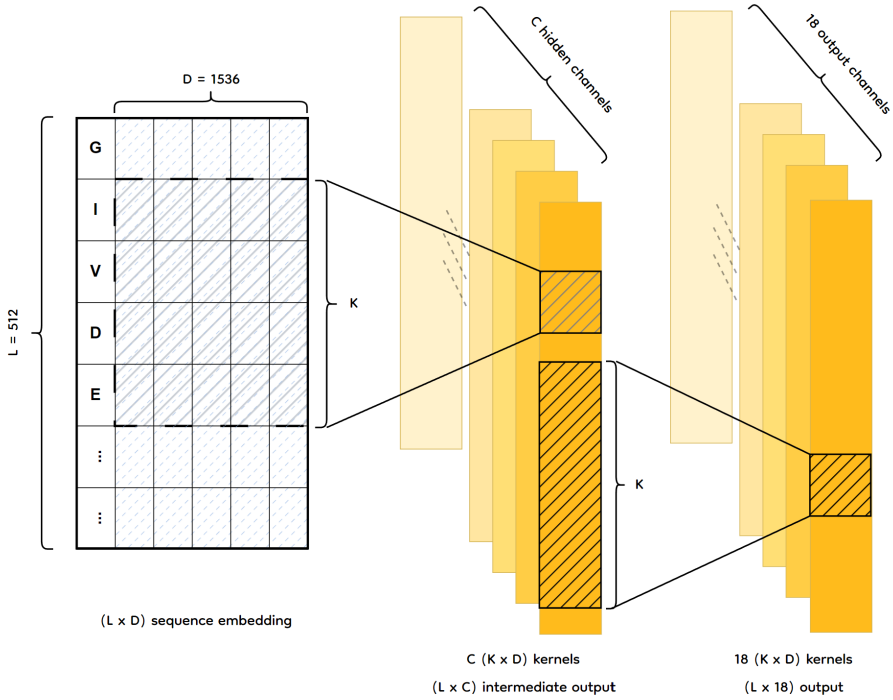


Figure 5.2: Two-layer 1D CNN architecture, taking Ankh embeddings as input.

5.3.2 Transformer Encoder

Over the past decade, transformer-based models have achieved remarkable success in natural language processing (NLP) domains. Proteins, often referred to as the "language of life", are composed of 20 amino acids that serve as their alphabet, making them well-suited for analysis using this NLP-inspired approach.

Differently from CNN, transformers utilize the self-attention mechanism, which enables them to capture long-range dependencies within protein sequences more effectively. Moreover, while CNNs require padding or truncating sequences to a fixed length, transformers can naturally work with sequences of different lengths.

Network architecture

As shown in Fig 5.3, the core component of the network is a stack of encoder layers inspired by [49], each comprises two sub-layers:

Multi-head Self-attention Layer:
A multi-head self-attention layer takes an input sequence representation matrix

$$\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, ..., \mathbf{a}_L]$$

where $\mathbf{a}_{i \in \{1,...,L\}}$ is the d-dimensional feature vector of the i-th amino acid in the sequence.

Given h attention heads, each head has three learnable weight matrices, i.e., $\mathbf{W}_q^{D' \times d_q}$, $\mathbf{W}_k^{D' \times d_k}$, and $\mathbf{W}_v^{D' \times d_v}$.

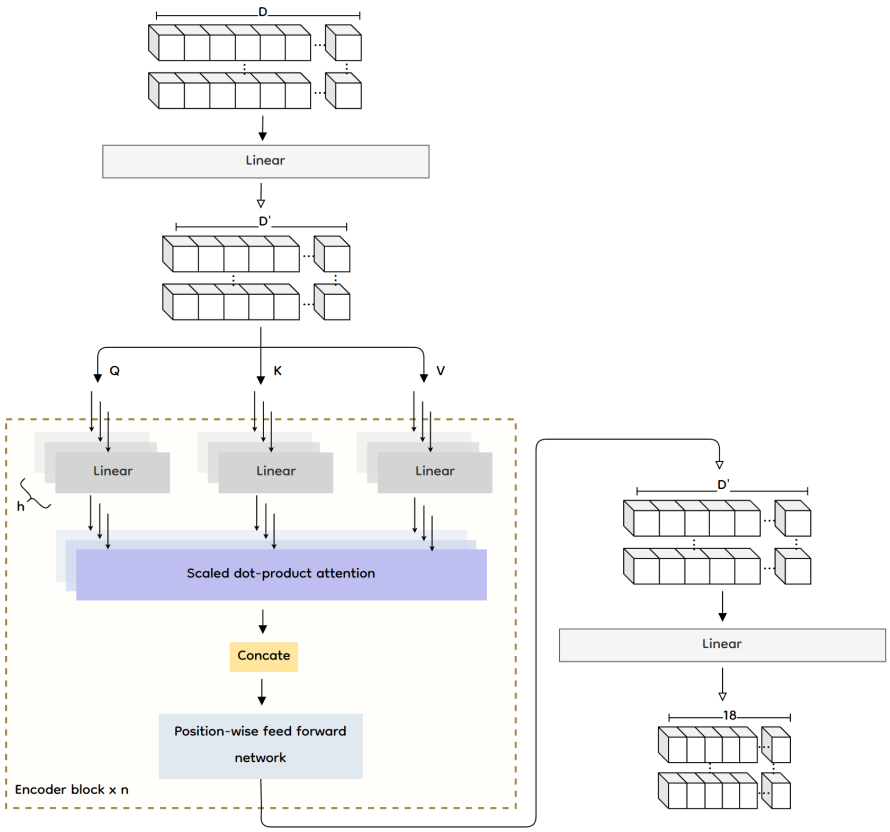


Figure 5.3: Transformer encoder network architecture.

The input vector $\mathbf{a}_i$ is projected by the three weight matrices to obtain its query, key, and value representations $\mathbf{q}_i$, $\mathbf{k}_i$, and $\mathbf{v}_i$, respectively:

$$\mathbf{q}_i = \mathbf{W}_q \mathbf{a}_i \tag{5.1}$$

$$\mathbf{k}_i = \mathbf{W}_k \mathbf{a}_i \tag{5.2}$$

$$\mathbf{v}_i = \mathbf{W}_v \mathbf{a}_i \tag{5.3}$$

The attention score of the j-th amino acid feature vector with respect to the target i-th amino acid is then calculated as

$$w_{ij} = \frac{\mathbf{q}_i \mathbf{k}_j}{\sqrt{d_k}} \tag{5.4}$$

The output of the attention layer captures the attention of the target i-th amino acid to all its peers in the sequence by

$$\mathbf{y}_i = \sum_j \text{softmax}(w_{ij}) \mathbf{v}_j \tag{5.5}$$

The $\frac{1}{\sqrt{d_k}}$ in (5.4) is the scaling factor to prevent large values to pass to the softmax function, where d_k is the dimension of the keys.

At the matrix manipulation level, the attention is obtained by

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathbf{softmax}(\frac{\mathbf{Q}\mathbf{K}^{\intercal}}{\sqrt{d_k}})\mathbf{V} \tag{5.6}$$

h self-attention heads are utilized to jointly attend to information from different representation subspaces, each with unique weight matrices of dimension $d_q = d_k = d_v = d/h$. The attention heads perform the attention function in parallel, and the outputs are concatenated and multiplied by an additional weights matrix $\mathbf{W}_o$, which captures information from all the attention heads:

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concate}(\text{head}_1, ..., \text{head}_h)\mathbf{W}_o \tag{5.7}$$

Positionwise Feed-Forward Layer:
Two linear transformations are applied to each position separately and identically, with a ReLU activation in between:

$$\text{FFL}(\mathbf{x}) = \max(0, \mathbf{x}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2 \tag{5.8}$$

This can be understood as two convolutions operated along the sequence with a kernel size of 1.

Multi-head self-attention introduces extra computational cost, to make the training more efficient, the input embeddings are projected to a lower hidden dimension, D', by a linear transformation before being fed to the transformer encoder layers.

Using the self-attention encoder, the query, key, and value are all generated from the input embeddings, both the sub-layers of the encoder perform a residual connection [70], followed by layer normalization [71].

All output dimensions of the encoder blocks are consistent with D', the hidden dimension of the position-wise feedforward network is $d_{ff} = 4D'$, following the same setting as in the paper "Attention is all you need" [49].

The outputs of the last encoder layer are fed to a dense layer followed by the final output layer of dimension 18, corresponding to the number of metal ion classes it aims to predict.

As a large-scale model, transformer is prone to overfitting. To cope with this issue, a dropout layer is inserted after each fully connected layer.

Hyperparameter

The transformer-based model involves several key hyperparameters:

- Hidden Dimension (D'): The hidden dimension, where embeddings are projected before entering encoder layers, impacts a model's capacity to learn complex protein sequence patterns. Larger dimensions boost capacity but may cause overfitting and longer training times.

- Number of Heads (h): The number of attention heads per encoder layer that allow the model to capture diverse relationships and dependencies in sequences. More heads can enhance learning and stabilize training [72].
- Number of Encoder Layers: Determines the depth of the model, with each layer capturing higher-level abstractions. More layers can improve the model's ability to learn complex patterns, but may lead to overfitting and longer training times.
- Dropout Rate: The probability of randomly deactivating neurons during training to prevent overfitting. Higher rates can improve generalization, but excessively high rates may lead to underfitting.

5.4 Model Training

5.4.1 Training Settings

Both models use a batch size of 16, an AdamW optimizer [73] of 1e-5 weigh decay, a cosine annealing learning rate scheduler [74] of $T_{\max} = 64$, a minimum epoch of 100 with early stopping observing no improvements in MCC and AUPRC for 30 epochs (the metrics are discussed in details in Section5.5).

The transformer model shows a more unstable training loss curve than the CNN model, therefore, a learning rate tuning is performed to find the optimal learning rate being around 0.000758. A grid search is also performed to fine-tune hyperparameters including the learning rates for both models, which is further discussed in Section 5.5.

5.4.2 Length Padding & Masking

Protein sequences are of various lengths, padding is necessary to make the embeddings appropriate for model training.

For the transformer model, shorter protein embeddings are zero-padded to meet the longest length in a batch. While for CNN, all proteins are padded to the pre-set maximum length of 512.

To correctly calculate the prediction loss of the padded proteins, a 1D one-zero mask is generated for each sequence embedding, implemented in the custom collate function of the `Dataset` module.

5.4.3 Loss Function

The Binary Cross-Entropy (BCE) loss function is employed to measure the divergence between the predicted probabilities and the ground truth labels. Specifically, the model output $\mathbf{o}$ is subjected to two transformations: masking the padded suffix and applying an element-wise Sigmoid function. The Sigmoid function maps each o_{im} to a probability value within the range $(0, 1)$:

$$p_{im} = \text{Sigmoid}[o_{im}] \tag{5.9}$$

Here, p_{im} denotes the probability of the i-th amino acid residue binding with metal ion m. The average BCE loss, normalized with respect to sequence length, is computed for each metal ion class, yielding 18 distinct losses. These losses are then combined for back-propagation:

$$L = \sum_{m=1}^{M} \frac{1}{N_m} \sum_{i=1}^{N_m} y_{im} \log(p_{im}) + (1 - y_{im}) \log(1 - p_{im}) \tag{5.10}$$

where $y_{im} \in \mathbf{y}$ represents the binding label of metal ion m for the i-th resiude.

5.4.4 Class Imbalance Cures

To cope with the severe class imbalance of the majority negative instances and minor positive instances, two approaches are used:

Weighting the loss function: To encourage the model to make more accurate and confident positive label predictions, the loss function is weighted to amplify the importance of positive labels. This can potentially lead to a simultaneous increase in both precision and recall. The weights are determined using the following formula:

$$\frac{W_1}{W_0} = \frac{\sqrt[n]{\#\text{neg}}}{\sqrt[n]{\#\text{pos}}} \tag{5.11}$$

The power n is a tunable hyperparameter. All metal ion classes share the same label weight to reduce the search space.

Downsampling negatives per epoch: In every training epoch, a subset of negative data (non-binding proteins) is selected and combined with all positive data (proteins with metal-binding sites). This approach increases the exposure of the model to positive instances, enabling it to better recognize and predict metal-binding sites.

5.5 Validation

5.5.1 Metrics

The following metrics are chosen for evaluating metal-binding site prediction models:

Precision (Positive Predictive Value):

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}} \tag{5.12}$$

Recall (Sensitivity, True Positive Rate):

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}} \tag{5.13}$$

Specificity (True Negative Rate):

$$\text{Specificity} = \frac{\text{True Negatives (TN)}}{\text{True Negatives (TN)} + \text{False Positives (FP)}} \tag{5.14}$$

Matthews Correlation Coefficient (MCC):

$$\text{MCC} = \frac{(\text{TP} \times \text{TN}) - (\text{FP} \times \text{FN})}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \tag{5.15}$$

AUROC (Area Under the Receiver Operating Characteristic curve)

AUPRC (Area Under the Precision-Recall Curve)

Both Precision and Recall are essential in evaluating the model's performance in predicting true positive cases (actual metal-binding sites). Notably, there is often a trade-off between the two; an aggressive positive-case predictor will yield high recall but low precision, while a conservative predictor will exhibit the opposite trend.

Specificity assesses the model's ability to accurately identify true negative cases (non-binding sites) among actual negative cases. Although this metric is typically high for most models due to the imbalanced nature of metal-binding site datasets, it is still included as negative data prediction also plays a role in model evaluation.

MCC is a crucial metric in this study as it provides a balanced measure, considering all four values of the confusion matrix (TP, TN, FP, and FN), and offers a single value that indicates the model's performance, accounting for both under- and over-predictions.

AUROC and AUPRC are probability-threshold-independent metrics. AUROC evaluates the model's ability to discriminate between positive and negative cases across varying thresholds, with a higher AUROC value indicating superior overall classification performance. AUPRC quantifies the trade-off between precision and recall. Consequently, a higher AUPRC value suggests a model that is more adept at handling imbalanced datasets.

5.5.2 Hyperparameter Tuning

Hyperparameter tuning serves the dual purpose of identifying the optimal hyperparameter set for the best model and enabling a comprehensive evaluation of the models' performance in a rigorous manner.

Both the CNN and transformer encoder models undergo hyperparameter optimization employing a five-fold grid search approach. In this process, the dataset is divided into five distinct splits, ensuring each subset contains binding samples for all metal ion classes. The AUPRC metric is specifically selected as the primary guiding criterion due to its sensitivity

to class imbalance, its ability to capture the trade-off between precision and recall, and its threshold-independent nature, which distinguishes it from metrics like MCC.

Tab 5.3 presents an overview of the pertinent hyperparameters and their respective search spaces for both models.

Table 5.3: Hyperparameter search spaces for the two-layer 1D CNN and transformer encoder models. [a] The first weight corresponds to label 0 and the second to label 1. [b] The pre-training-tuned learning rate.

Two-layer 1D CNN		**Transformer Encoder**	
Hyperparameter	**Search Space**	**Hyperparameter**	**Search Space**
Hidden Channel	[128, 256]	Hidden Dimension	[64, 128, 256]
Kernel Size	[13, 15, 17]	Number of Heads	[2, 4]
Learning Rate	[5e-4, 1e-3]	Number of Encoder Layers	[2, 3]
Loss Weights	[(0.22, 5.80), (0.78, 8.46), (0.39, 8.46), (0.69, 24.6)][a]	Dropout Rate	[0.1, 0.2, 0.3]
		Learning Rate	[0.0007586[b], 1e-3]
		Loss Weights	[(0.22, 5.80), (0.78, 8.46), (0.69, 24.6)][a]

5.5.3 Threshold Tuning

Metal-binding prediction is inherently probabilistic, and thus, setting an appropriate threshold is crucial for leveraging the full potential of the models.

To optimize the models' performance, a threshold tuning procedure that maximizes the Matthews Correlation Coefficient (MCC) is conducted for the best-performing models. This approach identifies the optimal metal-binding probability threshold for each metal ion class. The tuning is performed post-training on a single evaluation split, with a search space ranging from 0.10 to 0.99 at a step size of 0.01.

Chapter 6

Result Analysis

6.1 Analysis on Model Hyperparameter

The impact of different hyperparameters on model performance is analyzed based on the mean AUPRCs from the five-fold hyperparameter grid search.

Figure 6.1 and 6.2 illustrate the impact of hyperparameter choices on the performance of the transformer encoder and two-layer 1D CNN models, respectively.

For the transformer-based model, a general trend of performance improvement is observed with increasing hidden dimension size and number of attention heads. This aligns with the hypothesis that metal-binding site prediction depends on both the intrinsic properties of amino acids and their contextual environment. By increasing these two hyperparameters, the model is encouraged to learn more complex patterns over a broader scope.

Conversely, increasing the depth of the network does not guarantee better performance, as demonstrated in Figure 6.1c. As the network becomes deeper, the hidden features become more abstract, which may not contribute positively to metal-binding site prediction. Additionally, deeper networks are more prone to overfitting and require longer training times.

Regarding the dropout rate, an increase in this hyperparameter leads to an expansion of the 1.5 interquartile range (IQR), indicating that the model performance becomes less consistent. This inconsistency can be attributed to the inherent randomness of the dropout process. Models with a dropout rate of 0.1 demonstrate the most consistent evaluation results, while a rate of 0.2 raises the upper limit of model performance. A dropout rate of 0.3, however, may be excessive.

In terms of the learning rate, a slight performance advantage is observed when using the pre-tuned rate of 0.000758, which yields a marginally higher upper 1.5 IQR. The optimal label weighting ratio is found to be 0.78:8.46, which is generated by $\frac{\sqrt[3]{\#\text{Residue}_{pos}}}{\sqrt[3]{\#\text{Residue}_{neg}}}$.

Similar to the transformer-based model, the two-layer 1D CNN models exhibit a substantial increase in AUPRC when the hidden channel number is increased from 128 to 256. A higher kernel size also elevates the lower bound of AUPRC, with the largest kernel size of 17 resulting in the highest upper bound. The choice of learning rate significantly influences the model's performance. Models trained with a learning rate of 1e-3 are more likely to achieve better AUPRC compared to those with a learning rate of 5e-4, suggesting the latter may

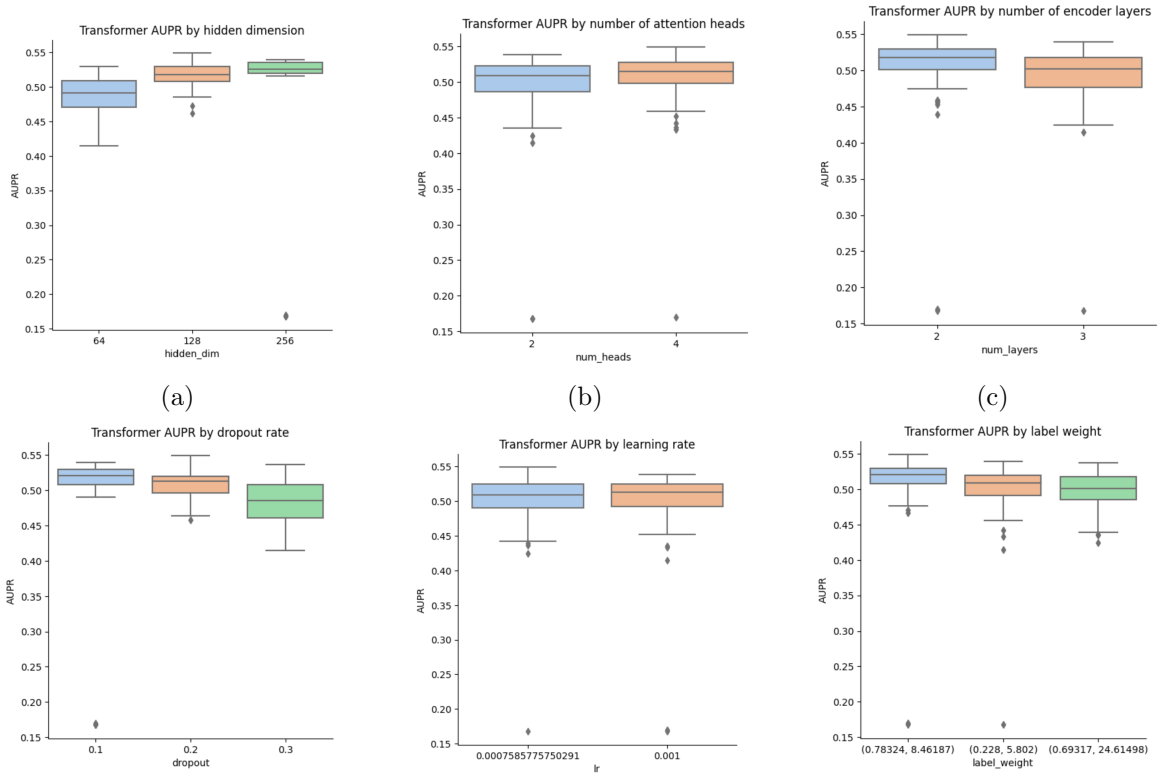


Figure 6.1: Hyperparameter impact on transformer encoder model performance reflected by the mean AUPRC. The box shows the interquatile range of the AUPRC of the models, highlighting the medium, the lower and upper 1.5*IQR, and any outliers.

lead to suboptimal convergence.

Regarding label weighting ratios, the ratios of 0.23:5.80, derived from $\frac{\sqrt[2]{\#\text{Protein}_{\text{metal-binding}}}}{\sqrt[2]{\#\text{Protein}_{\text{non-binding}}}}$, and 0.78:8.46, calculated as $\frac{\sqrt[3]{\#\text{Residue}_{pos}}}{\sqrt[3]{\#\text{Residue}_{neg}}}$, lead to the best-performing models. On the other hand, models using the ratio 0.39:8.46, which corresponds to $\frac{\sqrt[3]{\#\text{Residue}_{pos}\text{ per protein}}}{\sqrt[3]{\#\text{Residue}_{neg}\text{ per protein}}}$, exhibit a higher median and upper IQR.

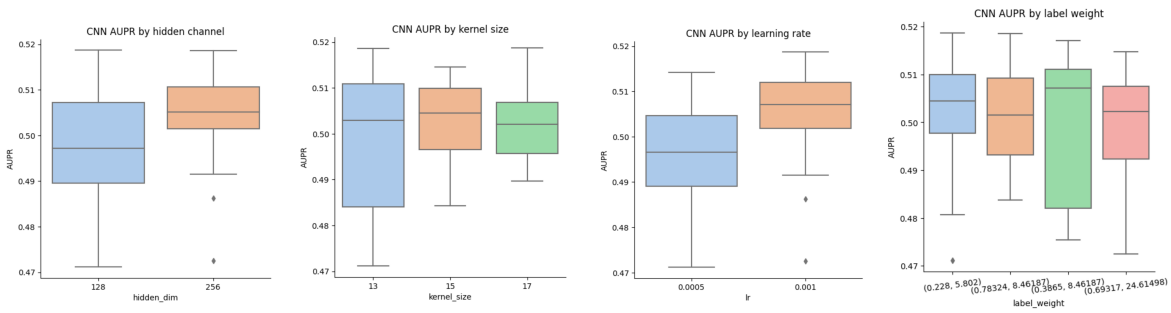


Figure 6.2: Hyperparameter impact on CNN model performance reflected by the mean AUPRC. The box shows the interquatile range of the AUPRC of the models, highlighting the medium, the lower and upper 1.5*IQR, and any outliers.

Upon comparing the performance of the transformer encoder models and the CNN models, it can be observed that the transformer models attain a higher upper bound AUPRC, with the 1.5 IQR nearing 0.55. In contrast, the CNN models demonstrate greater stability and consistency, as evidenced by a higher lower bound AUPRC of over 0.47 and fewer outliers.

6.2 Transformer Encoder versus CNN

6.2.1 Comparison of Prediction Results

To investigate the prediction performance of the two networks, the transformer encoder and two-layer 1D CNN models with the optimal hyperparameter combinations from the grid search are re-trained (detailed hyperparameter combinations shown in Tab 6.1). These models are referred to as **TFE** and **CNN2L**, respectively, throughout the remainder of this report. Their performances are evaluated on the independent test set, with the results compared in Tab 6.2.

Table 6.1: Best hyperparameter combinations found for CNN and transformer encoder models in terms of average AUPRC.

Two-layer 1D CNN		**Transformer Encoder**	
Hyperparameter	**Optimal Value**	**Hyperparameter**	**Optimal Value**
Hidden Channel	128	Hidden Dimension	128
Kernel Size	17	Number of Heads	4
Learning Rate	1e-3	Number of Encoder Layers	2
Loss Weights	(0.22, 5.80)	Dropout Rate	0.2
		Learning Rate	0.0007586
		Loss Weights	(0.78, 8.46)

In general, TFE surpasses CNN2L in terms of MCC scores at a probability threshold of 0.5 and AUPRC for all metal ion class predictions, except for Fe cation, where CNN2L attains a superior MCC with a 0.22 offset.

Consistent with the higher AUPRC, TFE typically outperforms CNN2L in both precision and recall, with the exception of Mg(2+) and Fe cation predictions, where CNN2L demonstrates an advantage in precision by 1.7% and 13.1%, respectively.

Interestingly, the highest AUROC values are usually achieved by CNN2L for all metal classes, except for Mn(2+), Ni(2+), Fe(2+), and Co(2+). Considering the TFE's advantage in AUPRC, the higher AUROC for CNN2L may be attributed to a lower false positive rate (FPR) as the true positive rate (TPR) varies. In this metal-binding site scenario, the FPR represents the number of errors made when predicting a non-binding site. Consequently, TFE appears to be a more aggressive metal-binding site predictor, while CNN2L is more conservative. The higher AUPRC and lower AUROC for TFE are expected, as the model

learns to predict metal-binding sites more effectively, it becomes more susceptible to making errors in those predictions.

The most notable prediction performance is achieved by the TFE model for iron-sulfur cluster families ([4Fe-4S] cluster, [2Fe-2S] cluster, and iron-sulfur cluster), Cu cation, and a metal cation, with the MCC scores and AUPRC exceeding 0.50. Interestingly, iron-sulfur cluster, Cu cation, and a metal cation are all parent metal classes, suggesting that the model may infer metal-binding status with the aid of the binding status of other metal ions it is predicting.

Conversely, both models exhibit poor predictive performance for several metal ions, including K(+), Ni(2+), Na(+), Fe(3+), Cu(2+), Fe(2+), and Co(2+), as evidenced by low precision, recall, MCC, and AUPRC. It is worth noting that these metal-binding samples are scarce in both the development set and the independent test set, with the highest being K(+) at only 20 protein samples. The lack of relevant data samples may cause both models to fail in generalizing. However, the situation is different for Mn(2+); despite the presence of 172 Mn(2+)-binding protein samples in the training set, both models still predict poorly for Mn(2+), with AUPRC of 0.080 and 0.022, respectively. A plausible explanation for the poor performance may be sought in the intrinsic biochemical properties of the Mn(2+)-binding sites, warranting further investigation to elucidate the underlying factors contributing to the difficulty in prediction.

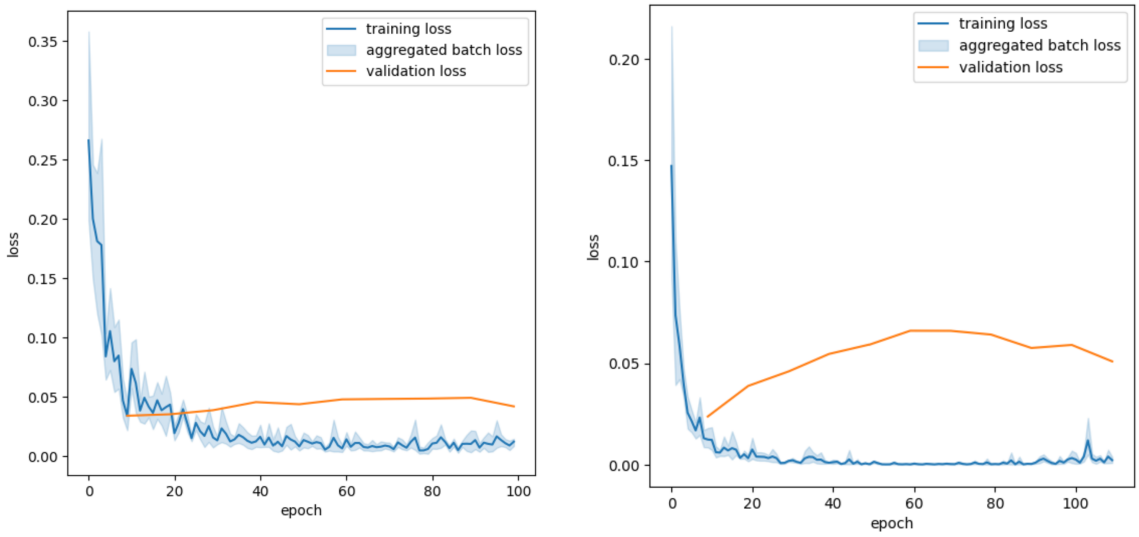
6.2.2 Discussion of Learning Curves

Upon examining the learning curves for both models depicted in Fig 6.3, several intriguing observations can be made. The validation loss curves for both models deviate from what is typically seen in other conventional machine learning tasks. This phenomenon can be attributed to the severe class imbalance inherent in metal-binding site prediction. Initially, the models are inclined to predict all instances as non-binding, resulting in low validation loss. However, guided by the label weights, the models gradually adapt to predict positive binding sites, causing the validation loss to increase due to a rise in false negatives. As the models become more accurate at predicting positive binding sites, the validation loss eventually exhibits the expected decrease.

Fig 6.4 displays the evolution of average AUPRC and average MCC at the probability threshold of 0.5 for both models on the validation set during training. Contrary to the validation loss, these two metrics rapidly improve as the models learn to predict positive binding sites. Both metrics reach their peak near epoch 80 and subsequently decline, particularly AUPRC, prompting early stopping of training around epoch 100. Throughout the training process, TFE consistently outperforms CNN2L in terms of AUPRC on the validation set. In terms of MCC, TFE surpasses CNN2L from around epoch 40 onwards, aligning with the results observed on the independent test set.

Table 6.2: The prediction performances of the hyperparameter-tuned transformer encoder model (TFE) and CNN model (CNN2L) evaluated on the test set. Precision, recall, specificity, and MCC are calculated using a threshold of 0.5. All metrics are rounded to three decimal places. Bold fonts denote the best results.

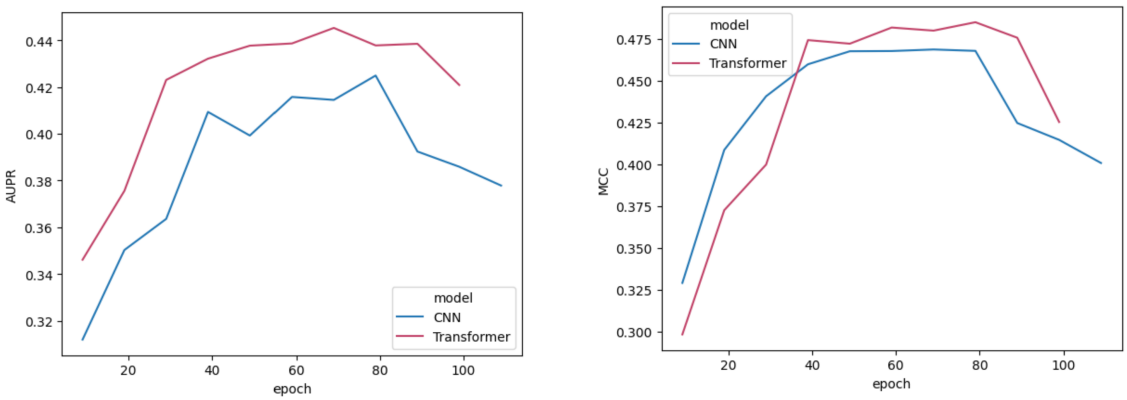
Metal Type	Predictor	Precision	Recall	Specificity	MCC	AUROC	AUPRC
Zn(2+)	TFE	**42.7**%	**65.3**%	100.0%	**0.528**	0.971	**0.529**
	CNN2L	33.9%	54.0%	100.0%	0.428	**0.985**	0.288
Mg(2+)	TFE	33.0%	**57.5**%	100.0%	**0.435**	0.945	**0.451**
	CNN2L	**34.7**%	41.2%	100.0%	0.378	**0.960**	0.301
[4Fe-4S] cluster	TFE	**40.5**%	**90.9**%	100.0%	**0.607**	1.000	**0.652**
	CNN2L	22.3%	66.7%	100.0%	0.386	1.000	0.155
Ca(2+)	TFE	**32.0**%	**52.3**%	100.0%	**0.409**	0.969	**0.340**
	CNN2L	22.3%	39.4%	100.0%	0.296	**0.979**	0.135
Mn(2+)	TFE	**12.7**%	**39.4**%	100.0%	**0.224**	**0.971**	**0.080**
	CNN2L	5.0%	9.9%	100.0%	0.070	0.956	0.022
a divalent metal cation	TFE	**39.3**%	**63.8**%	100.0%	**0.501**	0.960	**0.506**
	CNN2L	37.6%	50.9%	100.0%	0.437	**0.977**	0.316
Fe cation	TFE	35.5%	**38.6**%	100.0%	0.370	0.956	**0.226**
	CNN2L	**48.6**%	31.6%	100.0%	**0.392**	**0.978**	0.202
[2Fe-2S] cluster	TFE	**84.8**%	**96.6**%	100.0%	**0.905**	1.000	**0.966**
	CNN2L	75.0%	82.8%	100.0%	0.788	1.000	0.877
Cu cation	TFE	**72.3**%	**87.0**%	100.0%	**0.793**	0.959	**0.846**
	CNN2L	24.0%	75.9%	100.0%	0.427	**0.997**	0.145
K(+)	TFE	0.0%	0.0%	100.0%	0.000	0.445	0.000
	CNN2L	0.0%	0.0%	100.0%	0.000	**0.958**	**0.001**
Ni(2+)	TFE	0.0%	0.0%	100.0%	0.000	**0.999**	**0.002**
	CNN2L	0.0%	0.0%	100.0%	0.000	0.968	0.000
Na(+)	TFE	0.0%	0.0%	100.0%	0.000	0.831	0.000
	CNN2L	0.0%	0.0%	100.0%	0.000	**0.928**	0.000
Fe(3+)	TFE	0.0%	0.0%	100.0%	0.000	0.753	**0.001**
	CNN2L	0.0%	0.0%	100.0%	0.000	**0.847**	0.000
iron-sulfur cluster	TFE	**52.7**%	**92.5**%	100.0%	**0.698**	1.000	**0.831**
	CNN2L	38.5%	76.6%	100.0%	0.543	1.000	0.402
Cu(2+)	TFE	**16.7**%	**25.0**%	100.0%	**0.204**	1.000	**0.235**
	CNN2L	0.0%	0.0%	100.0%	0.000	1.000	0.007
Fe(2+)	TFE	0.0%	0.0%	100.0%	0.000	**0.999**	**0.001**
	CNN2L	0.0%	0.0%	100.0%	0.000	0.866	0.000
Co(2+)	TFE	0.0%	0.0%	100.0%	0.000	**0.886**	**0.003**
	CNN2L	0.0%	0.0%	100.0%	0.000	0.801	0.000
a metal cation	TFE	**41.9**%	**64.7**%	100.0%	**0.520**	0.935	**0.520**
	CNN2L	37.8%	52.9%	100.0%	0.447	**0.975**	0.343
Mean	TFE	**28.0**%	**43.0**%	100.0%	**0.344**	0.921	**0.344**
	CNN2L	21.1%	32.3%	100.0%	0.255	**0.954**	0.178



(a) TFE loss curves.

(b) CNN2L loss curves.

Figure 6.3: Training loss and validation loss over epoch. Batch losses aggregated per epoch with 95% confidence interval and mean as the epoch loss shown on the figures. Note that the the losses of the two models are not quantitatively comparable due to their different label weighting settings, however, the loss curve trends are comparable.



(a) AUPRC curves of TFE (in red) and CNN2L (in blue).

(b) MCC (thres=0.5) curves of TFE (in red) and CNN2L (in blue).

Figure 6.4: Comparison of metrics over epoch during training on validation set.

6.3 Training Longer No Better?

The learning curves of both the TFE and CNN2L models exhibit a declining validation loss immediately before early stopping based on average AUPRC. An experiment is conducted to explore the potential improvements if the models are trained for a longer duration.

Unfortunately, the CNN model demonstrates no improvement in AUPRC when trained for an additional 100 epochs. On the other hand, the transformer encoder model attains a new peak AUPRC on the validation set at epoch 209. This longer-trained model is referred to as **TFE-L** for the remainder of this report.

Fig 6.5 presents the learning curves of TFE-L, compared to the TFE model proposed in

Section 6.2. After the first 100 epochs, the training loss fluctuates, then steadily decreases and stabilizes after 150 epochs. This behavior could indicate that the model is exploring a new minimum to which it eventually converges. Concurrently, the validation loss experiences a surge from epoch 150 onwards and begins to decline toward the end of training.

Intriguingly, as illustrated by the average AUPRC and average MCC curves in Fig 6.6, an increase in validation loss coincides with a simultaneous rise in AUPRC and MCC. This phenomenon could result from the model learning new patterns and making more positive metal-binding predictions, thereby improving overall recall. However, these efforts also lead to a higher false positive rate, which negatively impacts the validation loss. Consequently, it is hypothesized that extending training for even longer epochs could yield similar fluctuations in validation loss and reciprocally related trends in AUPRC and MCC. However, the decision to stop training at epoch 209 is made based on the observed concrete convergence of the training loss.

The detailed prediction result comparison of the TFE and TFE-L models on the independent test set is presented in Appendix Fig A.1.

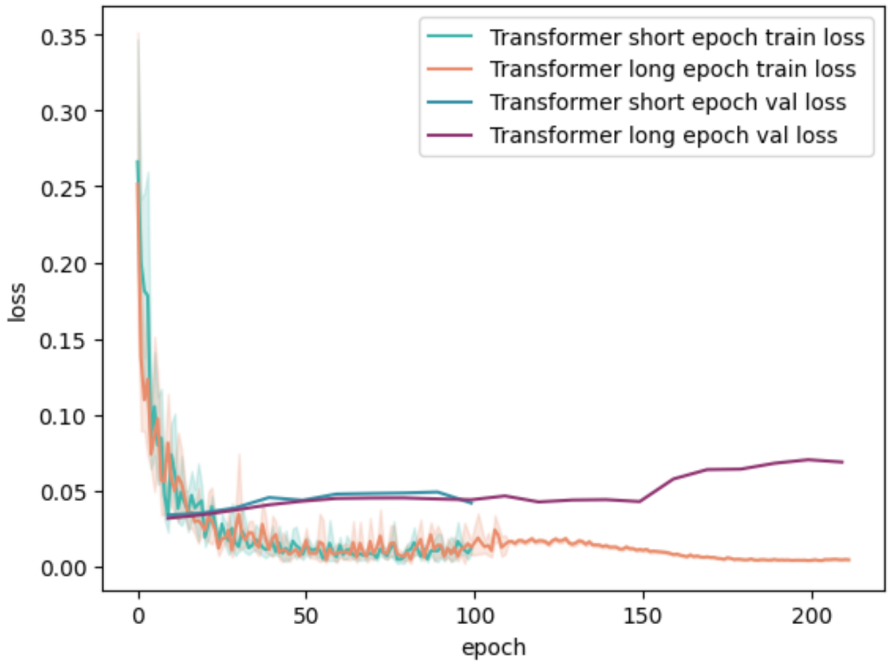
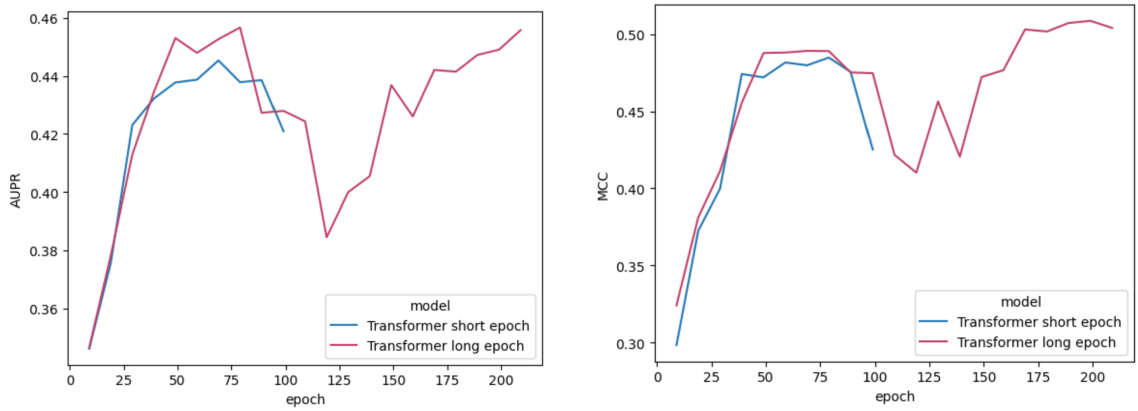


Figure 6.5: Training loss and validation loss curves of TFE (in cyan and in blue) and TFE-L (in orange and in scarlet).

6.4 Ankh versus ProtT5

An experimental comparison is conducted to evaluate the effectiveness of protein language model embeddings, specifically Ankh and ProtT5.

The transformer encoder model from Section 6.2 is retrained using ProtT5-XL-U50 generated embeddings, adhering to the same hyperparameter combination and training settings. This model is referred to as **TFE-T5** for the rest of the report.



(a) AUPRC curves of TFE (in blue) and TFE-L (in red).

(b) MCC (thres=0.5) curves of TFE (in blue) and TFE-L (in red).

Figure 6.6: Comparison of metrics over epoch for transformer encoder models trained for short epoch and long epoch

Upon comparing the prediction results on the independent test set, the model trained with Ankh embeddings generally outperforms the one using ProtT5, specifically, the average MCC and AUPRC across all metal classes are 0.344 and 0.344 using Ankh, and 0.287 and 0.316 using ProtT5 embeddings. However, ProtT5 embeddings demonstrate a better representation of Cu cation binding residues, with the model achieving a higher precision by 4%, MCC by 0.004, AUROC by 0.04, and AUPRC by 0.003, respectively.

It is noteworthy that, despite not being fine-tuned, TFE-T5 surpasses CNN2L in terms of both average MCC and average AUPRC by a significant margin of 0.032 and 0.138, respectively. This serves as compelling evidence of the transformer encoder network's superiority over the two-layer 1D CNN.

A detailed comparison of the prediction results for TFE and TFE-T5 on the independent test set can be found in Appendix Fig A.2.

6.5 Tuning the Probability Thresholds

To further boost model performance, probability thresholds are fine-tuned for each metal class, guided by average MCC on the validation set. The threshold-tuned models are referred to as **TFE-TFT** and **CNN2L-TFT** for the remainder of the report.

Tables 6.3 and 6.4 display the evaluation results of the TFE-TFT and CNN2L-TFT models compared to their corresponding untuned versions, respectively, on the independent test set.

As observed from the tables, the benefits of threshold fine-tuning vary significantly between the transformer encoder and two-layer 1D CNN models. Excluding metal ions that both models fail to recognize, such as K(+), Ni(2+), Na(+), Fe(3+), Fe(2+), and Co(2+), TFE-TFT experiences substantial performance gains in precision and MCC for all metal classes, except for Mn(2+). Specifically, precision and recall for Cu(2+) binding sites improve simultaneously by 3.3% and 25%.

In contrast, CNN2L-TFT exhibits a decline in MCC for nearly all metal types on the test set, except for a divalent metal cation, Fe cation, and a metal cation. Although there are precision gains, the improvements are much smaller than those observed for TFE-TFT.

Upon examining the specific metal predicting thresholds of the two models, it becomes apparent that the thresholds applied to the transformer encoder predictions are generally high. Among 18 metal labels, 13 have a threshold exceeding 0.90, indicating high confidence in positive binding-site predictions. Whereas, for the CNN model, thresholds appear more ambiguous, suggesting uncertainty in predictions.

It could be argued that if the thresholds fine-tuned on the validation set do not result in the anticipated performance improvements on the test set, which contains protein samples non-homologous to the development set, the model might suffer from prediction ambiguity and limited generalizability. This reasoning could also apply to the subpar Mn(2+)-binding prediction by TFE-TFT. A plausible hypothesis is that Mn(2+)-binding sites in the training set might share highly similar patterns with other metal-binding sites or exhibit a high degree of diversity in their patterns, leading to prediction failures on the test set.

Table 6.3: The performance of the transformer encoder model employing fine-tuned thresholds (TFE-TFT), with the changes in metrics (improvements or declines) following threshold fine-tuning specified within brackets.

Metal Type	Threshold	Precision	Recall	Specificity	MCC
Zn(2+)	0.93	52.1% (+9.4%)	57.1% (-8.2%)	100.0%	0.545 (+0.017)
Mg(2+)	0.96	51.4% (+18.4%)	46.3% (-11.2%)	100.0%	0.487 (+0.052)
[4Fe-4S] cluster	0.99	54.4% (+13.9%)	74.2% (-16.7%)	100.0%	0.636 (+0.029)
Ca(2+)	0.98	52.2% (+20.2%)	44.3% (-8.0%)	100.0%	0.481 (+0.072)
Mn(2+)	0.98	15.2% (+2.5%)	14.1% (-25.3%)	100.0%	0.146 (-0.078)
a divalent metal cation	0.99	64.1% (+24.8%)	47.3% (-16.5%)	100.0%	0.550 (+0.049)
Fe cation	0.98	51.2% (+15.7%)	36.8% (-1.8%)	100.0%	0.434 (+0.064)
[2Fe-2S] cluster	0.94	96.6% (+11.8%)	96.6% (–)	100.0%	0.966 (+0.061)
Cu cation	0.77	78.3% (+6.0%)	87.0% (–)	100.0%	0.826 (+0.033)
K(+)	0.21	0.0% (–)	0.0% (–)	100.0%	0.000 (–)
Ni(2+)	0.98	0.0% (–)	0.0% (–)	100.0%	0.000 (–)
Na(+)	0.96	0.0% (–)	0.0% (–)	100.0%	0.000 (–)
Fe(3+)	0.24	0.0% (–)	0.0% (–)	100.0%	0.000 (–)
iron-sulfur cluster	0.97	60.5% (+7.8%)	86.0% (-6.5%)	100.0%	0.721 (+0.023)
Cu(2+)	0.36	20.0% (+3.3%)	50.0% (+25.0%)	100.0%	0.316 (+0.112)
Fe(2+)	0.99	0.0% (–)	0.0% (–)	100.0%	0.000 (–)
Co(2+)	0.71	0.0% (–)	0.0% (–)	100.0%	0.000 (–)
a metal cation	0.99	65.5% (+23.6%)	47.9% (-16.8%)	100.0%	0.560 (+0.040)
Mean		36.7% (+8.7%)	38.2% (-4.8%)	100.0%	0.370 (+0.026)

Table 6.4: The performance of the two-layer 1D CNN model employing fine-tuned thresholds (CNN2L-TFT), with the changes in metrics (improvements or declines) following threshold fine-tuning specified within brackets.

Metal Type	Threshold	Precision	Recall	Specificity	MCC
Zn(2+)	0.99	42.4% (+8.5%)	41.2% (-12.8%)	100.0%	0.418 (-0.010)
Mg(2+)	0.59	35.0% (+0.3%)	40.0% (-1.2%)	100.0%	0.374 (-0.004)
[4Fe-4S] cluster	0.52	22.3% (–)	66.7% (–)	100.0%	0.386 (–)
Ca(2+)	0.80	24.0% (+1.7%)	35.2% (-4.2%)	100.0%	0.290 (-0.006)
Mn(2+)	0.76	3.4% (-1.6%)	5.6% (-4.3%)	100.0%	0.044 (-0.026)
a divalent metal cation	0.96	46.3% (+8.7%)	41.5% (-9.4%)	100.0%	0.438 (+0.001)
Fe cation	0.38	50.0% (+1.4%)	35.1% (+3.5%)	100.0%	0.419 (+0.027)
[2Fe-2S] cluster	0.74	78.6% (+3.6%)	75.9% (-6.9%)	100.0%	0.772 (-0.016)
Cu cation	0.26	23.2% (-0.8%)	77.8% (+1.9%)	100.0%	0.425 (-0.002)
K(+)	0.58	0.0% (–)	0.0% (–)	100.0%	0.000 (–)
Ni(2+)	0.30	0.0% (–)	0.0% (–)	100.0%	0.000 (–)
Na(+)	0.96	0.0% (–)	0.0% (–)	100.0%	0.000 (–)
Fe(3+)	0.10	0.0% (–)	0.0% (–)	100.0%	0.000 (–)
iron-sulfur cluster	0.97	42.1% (+3.6%)	67.3% (-9.3%)	100.0%	0.532 (-0.011)
Cu(2+)	0.36	0.0% (–)	0.0% (–)	100.0%	0.000 (–)
Fe(2+)	0.11	0.0% (–)	0.0% (–)	100.0%	0.000 (–)
Co(2+)	0.18	0.0% (–)	0.0% (–)	100.0%	0.000 (–)
a metal cation	0.85	43.1% (+5.3%)	48.3% (-4.6%)	100.0%	0.456 (+0.009)
Mean		22.8% (+1.7%)	29.7% (-2.6%)	100.0%	0.253 (-0.002)

6.6 Best Prediction Model from Best Predicted Metal Ions

Drawing on the evaluation results from Sections 6.1-6.5, this section presents the final version of the metal-binding prediction model developed in this study, referred to as **TFE-11**.

Taking into account the availability of data samples and the aforementioned experimental results, the model is designed to identify binding sites for 11 metal ion classes, including common protein-binding metal ions such as Zn(2+), Mg(2+), Ca(2+), as well as broader metal families such as Fe cation, Cu cation, iron-sulfur cluster, and a metal cation. It employs a transformer encoder network with the optimal hyperparameter combination identified in Section 6.1, utilizes an early stopping training strategy terminating at epoch 107, and undergoes threshold fine-tuning after training.

The model's performance is assessed on the independent test set, with evaluation results shown in Table 6.5. The model attains an average precision of 55.8%, recall of 56.4%, specificity of 100.0%, MCC of 0.558, AUROC of 0.960, and AUPRC of 0.558. It demonstrates exceptional performance in predicting all iron-sulfur cluster families and Cu cation, with the best result being [2Fe-2S] cluster, achieving 93.3% precision and 96.6% recall.

It is worth noting, however, that Mn(2+) prediction remains a challenge not only for this

model but also for other models throughout the experiments.

Table 6.5: Prediction performance of the TFE-11 model on the independent test set. All metrics round to three decimal places.

Metal Type	Precision	Recall	Specificity	MCC	AUROC	AUPRC
Zn(2+)	57.1%	49.4%	100.0%	0.531	0.970	0.513
Mg(2+)	56.2%	45.6%	100.0%	0.506	0.893	0.431
[4Fe-4S] cluster	56.1%	83.3%	100.0%	0.684	1.000	0.714
Ca(2+)	48.5%	42.8%	100.0%	0.456	0.976	0.357
Mn(2+)	4.90%	16.9%	100.0%	0.091	0.916	0.035
a divalent metal cation	56.2%	48.5%	100.0%	0.522	0.935	0.504
Fe cation	41.9%	31.6%	100.0%	0.364	0.980	0.233
[2Fe-2S] cluster	93.3%	96.6%	100.0%	0.949	0.999	0.950
Cu cation	75.0%	83.3%	100.0%	0.791	0.957	0.831
iron-sulfur cluster	65.7%	86.0%	100.0%	0.752	0.999	0.842
a metal cation	59.5%	49.3%	100.0%	0.541	0.933	0.521
Mean	55.8%	56.4%	100.0%	0.558	0.960	0.539

6.7 Comparison with Predictors from Related Work

In this section, the performance of the proposed TFE-11 model is evaluated by comparing it with metal-binding site predictors from other studies. It is important to note that the comparison is based on distinct independent test sets, due to the unavailability of specific tools and datasets. Furthermore, given the diversity of metal ion classes predicted by this model, identifying state-of-the-art methods for certain specific metal ion types is challenging. For example, while there are models for predicting Fe and Cu cation binding proteins, they do not identify specific binding sites, making them incomparable to this model.

Table 6.6 compares the TFE-11 model's performance with other predictors developed using well-established methodologies and comprehensively evaluated in their respective papers. For Ca(2+) and Mg(2+), TFE-11 achieves the highest recall among all predictors. For Zn(2+), TFE-11 outperforms most methods, with a recall only 8% lower than the highest. In the case of metal cations, TFE-11 significantly surpasses bindEmbed21 [37], a predictor utilizing ProtT5 embeddings, in both precision and recall. Compared to MetalPredator [24], which employs an MSA approach, this model maintains a recall advantage despite lower precision. The TFE-11 model exhibits perfect specificity (rounded to three significant digits) for all metal ion classes, emphasizing its effectiveness in accurately recognizing non-metal-binding sites. This aspect of the model's performance is crucial, as it reduces the rate of false positive predictions, potentially avoiding erroneous conclusions or unnecessary experiments.

In terms of MCC, TFE-11 shows the highest values for Ca(2+), Mg(2+), and the metal

cation category, indicating superior overall performance in these cases. For Zn(2+), although TFE-11 does not achieve the highest MCC, it still exhibits competitive performance. However, the prediction performance for Mn(2+) remains a significant weakness, with substantially lower values compared to other predictors in both recall and MCC score.

Table 6.6: Comparison of TFE-11 prediction performance with previous works on distinct independent test sets (due to the unavailability of the tools and datasets for certain methods). Bold fonts indicate the best results.

Metal Type	Predictor	Recall(%)	Specificity(%)	MCC
Zn(2+)	S-SITE[59]	56.4	98.2	0.379
	TargetS[27]	35.2	99.9	0.491
	IonSeq[26]	43.6	99.8	0.504
	COACH[59]	**57.4**	99.1	0.495
	IonCom[26]	48.9	99.9	**0.590**
	MIonSite[25]	46.1	99.8	0.557
	TFE-11	49.4	**100.0**	0.531
Ca(2+)	S-SITE	30.3	97.6	0.201
	TargetS	26.0	98.6	0.204
	IonSeq	22.7	99.0	0.211
	COACH	31.6	97.5	0.205
	IonCom	17.7	99.8	0.296
	MIonSite	15.0	99.7	0.232
	TFE-11	**42.8**	**100.0**	**0.456**
Mg(2+)	S-SITE	42.4	97.3	0.212
	TargetS	8.7	99.9	0.194
	IonSeq	5.6	100.0	0.183
	COACH	44.5	98.4	0.282
	IonCom	25.3	99.9	0.343
	MIonSite	13.0	99.9	0.257
	TFE-11	**45.6**	100.0	**0.506**
Mn(2+)	S-SITE	47.4	98.6	0.362
	TargetS	42.2	99.6	0.491
	IonSeq	31.1	99.8	0.455
	COACH	**54.4**	99.1	0.466
	IonCom	48.7	99.6	0.519
	MIonSite	47.0	99.7	**0.555**
	TFE-11	16.9	**100.0**	0.091
Metal Type	Predictor	Precision(%)	Recall(%)	MCC
a metal cation	bindEmbed21[37]	24.0	25.0	0.225
	TFE-11	**59.5**	**49.3**	**0.541**
iron–sulfur cluster	MetalPredator[24]	**89.1**	75.5	N.A.
	TFE-11	65.7	**86.0**	0.752

Chapter 7

Discussion

7.1 Discussion of the Methodologies

Protein language models (pLMs) offer novel opportunities for extracting information from proteins through a self-supervised, data-driven approach. By comparing the proposed model with state-of-the-art methods in Section 6.7, it becomes evident that the pLM embedding-based model developed in this project exhibits competitive performance, challenging predictors that rely on homology inference, multiple sequence alignment (MSA), or traditional protein structural feature representations.

In terms of network architectures, both the transformer encoder and two-layer 1D CNN have demonstrated their capacity to effectively learn from pLM embeddings and generalize to varying extents. This is supported by the models' ability to perform well on a test set consisting of proteins with low similarity to those in the development set, despite being trained with homologous proteins.

When comparing the two network designs, the transformer encoder model holds a significant advantage over the CNN model in almost all metric measurements. As discussed in Section 3.1, the local environment of an amino acid residue is a critical factor in determining its metal-binding property. However, defining the scope of these "local environments" is challenging, both heuristically and theoretically, and the protein primary structure does not fully represent the spatial geometry of the residues. This is where the self-attention layer's ability to capture global information becomes crucial. Unlike convolutional neural networks, which have their scope constrained by kernel size, the self-attention mechanism in transformer encoder models is not limited in this way. The attention mechanism, as suggested in [75], incorporates not only a subspace of the states it attends to but also affine transformations of those states that preserve nearly their entire information content. This allows the attention mechanism to route multiple local information sources into a unified global tree structure of local representations, demonstrating the superiority of the transformer encoder network for this specific task.

Importantly, compared to CNNs, employing a transformer encoder network necessitates more meticulous hyperparameter fine-tuning to maximize its performance. This is evidenced by the greater performance variance observed in the hyperparameter grid search of the

transformer encoder network conducted in this study.

7.2 Discussion of the Results

7.2.1 Low Precision High Recall Issue

The prediction models developed in this study tend to exhibit low precision and high recall. From a machine learning standpoint, this is a common challenge in class imbalance tasks. With positive (binding) samples representing the minority class, a large number of negative (non-binding) examples may be misclassified as false positives. Conversely, the scarcity of positive instances leads to fewer false negatives, resulting in low precision and high recall. By utilizing class imbalance mitigation techniques, as detailed in Section 5.4.4, this issue has been effectively addressed, improving the models' overall performance. From a biological perspective, residues not observed as binding are assumed to be true non-binding sites in the utilized datasets. However, limitations in experimental annotations constrain the maximum attainable precision.

7.2.2 High Specificity High Accuracy $\neq$ Good Model

With the presence of class imbalance in the dataset, high specificity and high accuracy do not necessarily imply a good metal-binding site prediction model.

High specificity indicates that the model correctly classifies a large proportion of the negative (non-binding) examples. However, it does not provide information about the model's performance on the positive (binding) classes, which is the primary objective in predicting metal-binding sites. A model could achieve high specificity and high accuracy by simply classifying most instances as negative, especially when the negative class is predominant.

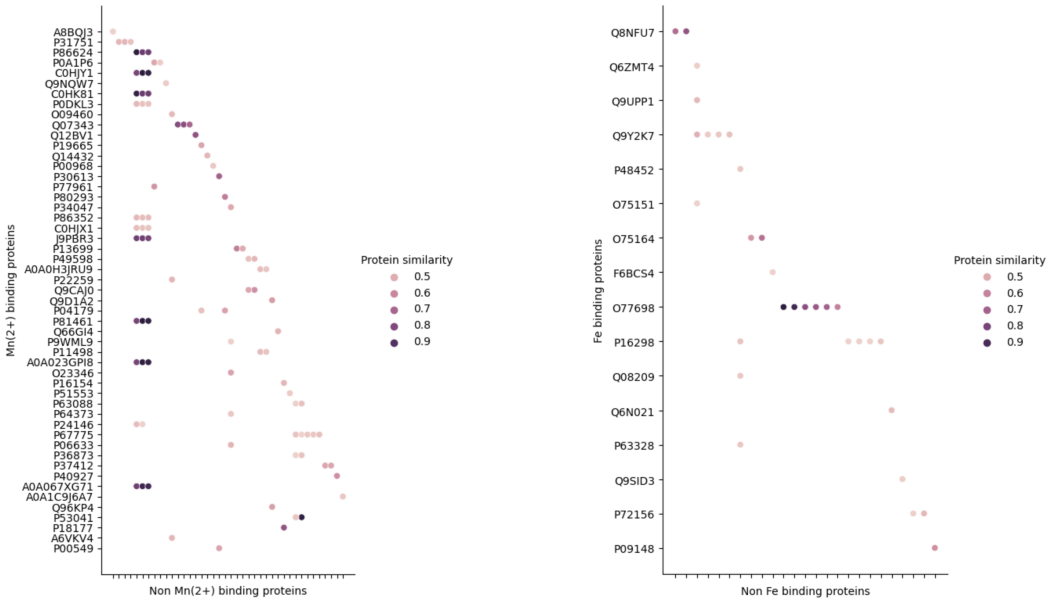
To develop models suitable for practical applications, it is essential to emphasize the mutual improvement of precision and recall, as captured by MCC, F-score, and AUPRC. These metrics provide a comprehensive assessment of the model's ability to accurately identify both binding and non-binding examples, ultimately reflecting the true performance of the model in predicting metal-binding sites.

7.2.3 Why Mn(2+)-Binding Site is Especially Hard

This analysis aims to explain the poor prediction performance for Mn(2+)-binding sites, particularly from a data science perspective. To investigate this, Mn(2+)-binding proteins and Fe cation binding proteins in the datasets are compared.

As illustrated in Fig 7.1, 51 out of 195 (26%) Mn(2+)-binding proteins share above 40% pairwise similarity with non-Mn(2+)-binding proteins. In contrast, for Fe cation binding proteins, only 16 out of 170 (9.4%) exhibit such similarity. Considering also that Mn(2+) shares similar ionic properties with other divalent metal ions, such as Mg(2+), Ca(2+), and Fe(2+), it becomes challenging to distinguish Mn(2+)-binding sites from those of other metal ions. This similarity may contribute to an increased rate of false positives and false negatives in the prediction model.

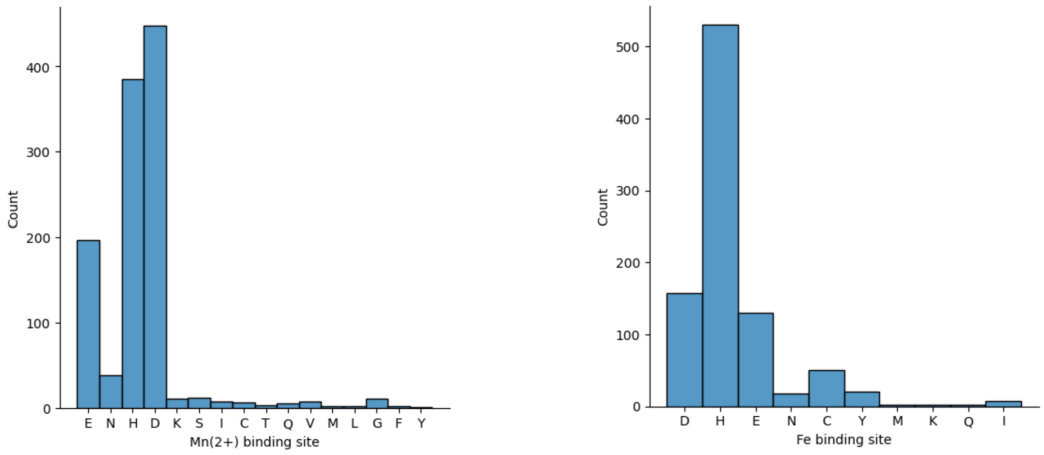
Furthermore, as shown in Fig 7.2, Mn(2+) ions exhibit a capacity to bind to a diverse set of amino acids, specifically 16 distinct amino acids, while having concurrently high propensities to bind to E, H, and D. In contrast, Fe cation binding residues demonstrate a more conservative profile, binding to merely 10 amino acids in the utilized datasets, and showing a uniquely high affinity for the amino acid H. This extensive diversity of Mn(2+)-binding sites poses a challenge in identifying conserved structural features for these sites, further complicating their prediction.



(a) The pairwise protein similarity (> 40%) of Mn(2+)-binding proteins and non Mn(2+)-binding proteins

(b) The pairwise protein similarity (> 40%) of Fe cation binding proteins and non Fe binding proteins

Figure 7.1: Comparison of Mn(2+) and Fe cation binding proteins in terms of the similar corresponding non-binding proteins.



(a) The Mn(2+)-binding amino acid frequency.

(b) The Fe cation binding amino acid frequency.

Figure 7.2: Comparison of Mn(2+) and Fe cation binding amino acid frequencies.

7.2.4 Biological Significance of the Model

The metal-binding site prediction model presented in this project holds significant biological implications, as it offers a powerful tool for researchers to investigate the diverse roles of metal-binding sites in proteins. By providing adjustable threshold settings, the model allows for predictions with varying confidence levels, which can be tailored to meet the specific objectives and requirements of individual research projects.

When employing a high threshold, the model yields fewer but more confident predictions, making it suitable for validating existing hypotheses regarding metal-binding sites. On the other hand, utilizing a lower threshold generates more inclusive predictions, enabling researchers to explore and potentially discover new metal-binding sites that may have been previously overlooked. By offering a broader range of potential metal-binding sites, the model can facilitate the planning and execution of laboratory experiments aimed at investigating the role of metal ions in protein systems. This can, in turn, lead to the elucidation of novel molecular mechanisms, the identification of new drug targets, and the development of groundbreaking therapeutic approaches.

Chapter 8

Conclusions

8.1 Achievements

This dissertation project yields several noteworthy outcomes. First, it presents a comprehensive literature review of metal-binding site prediction studies, assembling the state-of-the-art approaches and findings in the field. This foundation facilitates a thorough understanding of the current landscape and informs the development of novel prediction models.

Secondly, the project develops a robust model, TFE-11, which accurately predicts the binding sites of 11 metal ion classes. The model demonstrates competitive or improved performance compared to existing methods, achieving an average MCC of 0.558, AUROC of 0.96, and AUPRC of 0.539. This achievement showcases the effectiveness of the proposed approach and its potential to contribute to the field of metal-binding site prediction.

Furthermore, the project provides insights into the methodologies appropriate for metal-binding site prediction tasks. This includes aspects such as label and feature representation, network selection, and equitable evaluation of model performance. These insights impart valuable knowledge to the research community and help guide future work in this area.

Additionally, the project investigates the application of protein language model (pLM) embeddings for supervised downstream tasks. The proposed model is the first to utilize Ankh embeddings specifically for metal-binding site prediction tasks, highlighting the potential of employing pLM embeddings in solving biologically relevant problems.

Lastly, the project provides open-source codes (with the GitHub repository link and main tools utilized in Appx. B) for the model development pipelines that can be reused and adapted to accommodate more metal ion classes by other researchers. This facilitates the adoption and extension of the proposed approach for a wider range of applications.

8.2 Limitations & Future Work

The metal-binding site prediction model presented in this study has demonstrated promising results. However, there are certain limitations and areas for future improvement that need to be addressed:

Data insufficiency: The transformer model requires a large amount of data to per-

form optimally. Manually annotating data from protein databases, such as PDB, is time-consuming and requires significant biological knowledge and tools. Consequently, only datasets provided by the UniProt challenges were utilized in this study. In the future, more efficient strategies for data annotation or data augmentation techniques could be employed to generate a larger training set.

Incorporating 3D structure: The current model does not consider the 3D structure of proteins in the prediction process. Future work could focus on incorporating structural information to improve the accuracy of predictions.

Fine-tuning protein language model: Fine-tuning the protein language model to make it more suitable for this specific task could enhance the model's ability to capture relevant features and, consequently, improve the overall performance of the metal-binding site prediction.

Improving the model design: Enhancements can be made to both the design and training of the transformer encoder and CNN models, including incorporating diverse features, adopting more sophisticated strategies to manage exceedingly long input sequences, assigning specific weights to individual metal ion classes, and utilizing active learning and multitask learning strategies for more effective learning.

Exploring alternative models: The use of alternative architectures, such as ConvBERT, which has been proven to excel in downstream tasks in Ankh's paper [67], could potentially provide better results and should be explored in future research.

Ensemble networks: Some models were observed to excel at binding sites for specific metal ions. Combining these specialized models into an ensemble network could lead to improved performance across different metal ion classes.

Robust evaluation and comparison with peer works: When comparing the model performance with other approaches, more robust evaluation strategies should be employed, such as testing on blind datasets. Additionally, promoting open-source tools would facilitate transparent and fair comparisons between different methods.

Bibliography

[1] E. A. Permyakov, "Metal binding proteins," *Encyclopedia*, vol. 1, no. 1, pp. 261–292, Mar. 2021. [Online]. Available: https://www.mdpi.com/2673-8392/1/1/24

[2] T. U. Consortium, "UniProt: the universal protein knowledgebase in 2021," *Nucleic Acids Research*, vol. 49, no. D1, pp. D480–D489, 11 2020. [Online]. Available: https://doi.org/10.1093/nar/gkaa1100

[3] Uniprot, "Inside uniprot: the uniprot metal binding site machine learning challenge," Feb. 2022. [Online]. Available: https://insideuniprot.blogspot.com/2022/02/the-uniprot-metal-binding-site-machine.html

[4] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, "Protein function," *Molecular Biology of the Cell. 4th edition*, 2002. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK26911/

[5] T. Sanvictores and F. Farci, "Biochemistry, primary protein structure," Oct 2022. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK564343/

[6] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, "The shape and structure of proteins," *Molecular Biology of the Cell. 4th edition*, 2002. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK26830/

[7] P. Lian, R. C. Johnston, J. M. Parks, and J. C. Smith, "Quantum chemical calculation of pkas of environmentally relevant functional groups: Carboxylic acids, amines, and thiols in aqueous solution," *The Journal of Physical Chemistry A*, vol. 122, no. 17, pp. 4366–4374, 2018, pMID: 29633840. [Online]. Available: https://doi.org/10.1021/acs.jpca.8b01751

[8] K. Steward, "Amino acids – the building blocks of proteins," Sep 2019. [Online]. Available: https://www.technologynetworks.com/applied-sciences/articles/essential-amino-acids-chart-abbreviations-and-structure-324357

[9] M. Weiss, D. F. Steiner, and L. H. Philipson, "Insulin biosynthesis, secretion, structure, and structure-activity relationships," in *Endotext*, K. R. Feingold, B. Anawalt, M. R. Blackman, A. Boyce, G. Chrousos, E. Corpas, W. W. de Herder, K. Dhatariya, K. Dungan, J. Hofland, S. Kalra, G. Kaltsas, N. Kapoor, C. Koch, P. Kopp, M. Korbonits, C. S. Kovacs, W. Kuohung, B. Laferrère,

M. Levy, E. A. McGee, R. McLachlan, M. New, J. Purnell, R. Sahay, A. S. Shah, F. Singer, M. A. Sperling, C. A. Stratakis, D. L. Trence, and D. P. Wilson, Eds. South Dartmouth (MA): MDText.com, Inc., 2000. [Online]. Available: http://www.ncbi.nlm.nih.gov/books/NBK279029/

[10] K. D. Berndt, "Types of Secondary Structure," 1996, publisher: Karolinska Institute, Stockholm, Sweden. [Online]. Available: https://www.cryst.bbk.ac.uk/PPS2/course/section8/ss-960531_4.html

[11] P. J. Worsfold, A. Townshend, and C. F. Poole, *Encyclopedia of analytical science*. S.l.: Elsevier, 2010.

[12] D. W. Christianson, "Structural chemistry and biology of manganese metalloenzymes," *Progress in Biophysics and Molecular Biology*, vol. 67, no. 2, pp. 217–252, 1997. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0079610797884775

[13] R. R. Mendel, A. G. Smith, A. Marquet, and M. J. Warren, "Metal and cofactor insertion," *Natural Product Reports*, vol. 24, no. 5, p. 963, 2007. [Online]. Available: http://xlink.rsc.org/?DOI=b703112m

[14] A. Rodzik, P. Pomastowski, G. N. Sagandykova, and B. Buszewski, "Interactions of whey proteins with metal ions," *International Journal of Molecular Sciences*, vol. 21, no. 6, p. 2156, Mar. 2020. [Online]. Available: https://www.mdpi.com/1422-0067/21/6/2156

[15] C.-T. LIN, K.-L. LIN, C.-H. YANG, I.-F. CHUNG, C.-D. HUANG, and Y.-S. YANG, "Protein metal binding residue prediction based on neural networks," *International Journal of Neural Systems*, vol. 15, no. 01n02, pp. 71–84, 2005, pMID: 15912584. [Online]. Available: https://doi.org/10.1142/S0129065705000116

[16] J. A. Tainer, V. A. Roberts, and E. D. Getzoff, "Protein metal-binding sites," *Current Opinion in Biotechnology*, vol. 3, no. 4, pp. 378–387, 1992. [Online]. Available: https://www.sciencedirect.com/science/article/pii/095816699290166G

[17] L. Riccardi, V. Genna, and M. De Vivo, "Metal–ligand interactions in drug design," *Nature Reviews Chemistry*, vol. 2, no. 7, pp. 100–112, Jun. 2018. [Online]. Available: https://www.nature.com/articles/s41570-018-0018-6

[18] E. Di Cera, "Mechanisms of ligand binding," *Biophysics Reviews*, vol. 1, no. 1, p. 011303, Dec. 2020. [Online]. Available: https://pubs.aip.org/aip/bpr/article/134313

[19] M. Hendlich, F. Rippmann, and G. Barnickel, "Ligsite: automatic and efficient detection of potential small molecule-binding sites in proteins," *Journal of Molecular Graphics and Modelling*, vol. 15, no. 6, pp. 359–363, 1997. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1093326398000023

[20] D. G. Levitt and L. J. Banaszak, "Pocket: A computer graphies method for identifying and displaying protein cavities and their surrounding amino acids," *Journal of Molecular Graphics*, vol. 10, no. 4, pp. 229–234, 1992. [Online]. Available: https://www.sciencedirect.com/science/article/pii/026378559280074N

[21] M. Hernandez, D. Ghersi, and R. Sanchez, "SITEHOUND-web: a server for ligand binding site identification in protein structures," *Nucleic Acids Research*, vol. 37, no. suppl_2, pp. W413–W416, 04 2009. [Online]. Available: https://doi.org/10.1093/nar/gkp281

[22] A. T. R. Laurie and R. M. Jackson, "Q-SiteFinder: an energy-based method for the prediction of protein–ligand binding sites," *Bioinformatics*, vol. 21, no. 9, pp. 1908–1916, 02 2005. [Online]. Available: https://doi.org/10.1093/bioinformatics/bti315

[23] Y.-F. Lin, C.-W. Cheng, C.-S. Shih, J.-K. Hwang, C.-S. Yu, and C.-H. Lu, "Mib: Metal ion-binding site prediction and docking server," *Journal of Chemical Information and Modeling*, vol. 56, no. 12, pp. 2287–2291, 2016, pMID: 27976886. [Online]. Available: https://doi.org/10.1021/acs.jcim.6b00407

[24] Y. Valasatava, A. Rosato, L. Banci, and C. Andreini, "MetalPredator: a web server to predict iron–sulfur cluster binding proteomes," *Bioinformatics*, vol. 32, no. 18, pp. 2850–2852, 06 2016. [Online]. Available: https://doi.org/10.1093/bioinformatics/btw238

[25] L. Qiao and D. Xie, "Mionsite: Ligand-specific prediction of metal ion-binding sites via enhanced adaboost algorithm with protein sequence information," *Analytical Biochemistry*, vol. 566, pp. 75–88, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0003269718304925

[26] X. Hu, Q. Dong, J. Yang, and Y. Zhang, " Recognizing metal and acid radical ion-binding sites by integrating ab initio modeling with template-based transferals ," *Bioinformatics*, vol. 32, no. 21, pp. 3260–3269, 07 2016. [Online]. Available: https://doi.org/10.1093/bioinformatics/btw396

[27] D.-J. Yu, J. Hu, J. Yang, H.-B. Shen, J. Tang, and J.-Y. Yang, "Designing template-free predictor for targeting protein-ligand binding sites with classifier ensemble and spatial clustering," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 10, no. 4, pp. 994–1008, 2013.

[28] S. Kumar, "Prediction of metal ion binding sites in proteins from amino acid sequences by using simplified amino acid alphabets and random forest model," *Genomics Inform*, vol. 15, no. 4, p. 162–169, 2017.

[29] L. Liu, X. Hu, Z. Feng, S. Wang, K. Sun, and S. Xu, "Recognizing ion ligand–binding residues by random forest algorithm based on optimized dihedral angle," *Frontiers in Bioengineering and Biotechnology*, vol. 8, 2020. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fbioe.2020.00493

[30] J. Song, C. Li, C. Zheng, J. Revote, Z. Zhang, and G. Webb, "Metalexplorer, a bioinformatics tool for the improved prediction of eight types of metal-binding sites using a random forest algorithm with two-step feature selection," *Current Bioinformatics*, vol. 11, pp. 1–1, 06 2016.

[31] X. Hu, Z. Feng, X. Zhang, L. Liu, and S. Wang, "The identification of metal ion ligand-binding residues by adding the reclassified relative solvent accessibility," *Frontiers in Genetics*, vol. 11, 2020. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fgene.2020.00214

[32] Z. Zhao, Y. Xu, and Y. Zhao, "Sxgbsite: Prediction of protein–ligand binding sites using sequence information and extreme gradient boosting," *Genes*, vol. 10, no. 12, 2019. [Online]. Available: https://www.mdpi.com/2073-4425/10/12/965

[33] Y. Ding, J. Tang, and F. Guo, "Identification of protein–ligand binding sites by sequence information and ensemble classifier," *Journal of Chemical Information and Modeling*, vol. 57, no. 12, pp. 3149–3161, 2017, pMID: 29125297. [Online]. Available: https://doi.org/10.1021/acs.jcim.7b00307

[34] Haberal and H. Oğul, "Prediction of protein metal binding sites using deep neural networks," *Molecular Informatics*, vol. 38, no. 7, p. 1800169, Jul. 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/minf.201800169

[35] A. Aptekmann, J. Buongiorno, D. Giovannelli, M. Glamoclija, D. Ferreiro, and Y. Bromberg, "mebipred: identifying metal-binding potential in protein sequence," *bioRxiv*, 2021. [Online]. Available: https://www.biorxiv.org/content/early/2021/08/13/2021.08.12.456141

[36] C.-Q. Xia, X. Pan, and H.-B. Shen, "Protein–ligand binding residue prediction enhancement through hybrid deep heterogeneous learning of sequence and structure data," *Bioinformatics*, vol. 36, no. 10, pp. 3018–3027, 02 2020. [Online]. Available: https://doi.org/10.1093/bioinformatics/btaa110

[37] M. Littmann, M. Heinzinger, C. Dallago, K. Weissenow, and B. Rost, "Protein embeddings and deep learning predict binding residues for various ligand classes," *Scientific Reports*, vol. 11, no. 1, p. 23916, Dec. 2021. [Online]. Available: https://www.nature.com/articles/s41598-021-03431-4

[38] Q. Yuan, S. Chen, Y. Wang, H. Zhao, and Y. Yang, "Alignment-free metal ion-binding site prediction from protein sequence through pretrained language model and multi-task learning," *Briefings in Bioinformatics*, vol. 23, no. 6, 10 2022, bbac444. [Online]. Available: https://doi.org/10.1093/bib/bbac444

[39] T. Olenyi, C. Marquet, M. Heinzinger, B. Kröger, T. Nikolova, M. Bernhofer, P. Sändig, K. Schütze, M. Littmann, M. Mirdita, M. Steinegger, C. Dallago, and B. Rost, "Lambdapp: Fast and accessible protein-specific phenotype predictions,"

bioRxiv, 2022. [Online]. Available: https://www.biorxiv.org/content/early/2022/08/05/2022.08.04.502750

[40] Y. Cheng, H. Wang, H. Xu, Y. Liu, B. Ma, X. Chen, X. Zeng, X. Wang, B. Wang, C. Shiau, S. Ovchinnikov, X.-D. Su, and C. Wang, "Co-evolution-based prediction of metal-binding sites in proteomes by machine learning," *Nature Chemical Biology*, Jan. 2023. [Online]. Available: https://www.nature.com/articles/s41589-022-01223-z

[41] A. Elnaggar, M. Heinzinger, C. Dallago, G. Rehawi, Y. Wang, L. Jones, T. Gibbs, T. Feher, C. Angerer, M. Steinegger, D. Bhowmik, and B. Rost, "Prottrans: Toward understanding the language of life through self-supervised learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 7112–7127, 2022.

[42] N. S. Detlefsen, S. Hauberg, and W. Boomsma, "Learning meaningful representations of protein sequences," *Nature Communications*, vol. 13, no. 1, 2022.

[43] A. Mohammadi, J. Zahiri, S. Mohammadi, M. Khodarahmi, and S. S. Arab, "PSSMCOOL: a comprehensive R package for generating evolutionary-based descriptors of protein sequences from PSSM profiles," *Biology Methods and Protocols*, vol. 7, no. 1, 03 2022, bpac008. [Online]. Available: https://doi.org/10.1093/biomethods/bpac008

[44] M. H. Kang, H. Y. Yoon, and Y. W. Choi, "Chapter 5 - ripl peptide as a novel cell-penetrating and homing peptide: design, characterization, and application to liposomal nanocarriers for hepsin-specific intracellular drug delivery," in *Nanostructures for Cancer Therapy*, ser. Micro and Nano Technologies, A. Ficai and A. M. Grumezescu, Eds. Elsevier, 2017, pp. 129–157. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780323461443000052

[45] M. Z. Tien, A. G. Meyer, D. K. Sydykova, S. J. Spielman, and C. O. Wilke, "Maximum allowed solvent accessibilites of residues in proteins," *PLOS ONE*, vol. 8, no. 11, pp. 1–8, 11 2013. [Online]. Available: https://doi.org/10.1371/journal.pone.0080635

[46] "Dihedral angles." [Online]. Available: http://www.mlb.co.jp/linux/science/garlic/doc/commands/dihedrals.html#:~:text=MAIN%20CHAIN%20DIHEDRAL%20ANGLES&text=(1)%20The%20phi%20angle%20is,cis%20to%20N%2DCA%20bond.

[47] C. Guo, M. Cheng, and M. L. Gross, "Protein-metal-ion interactions studied by mass spectrometry-based footprinting with isotope-encoded benzhydrazide," *Analytical Chemistry*, vol. 91, no. 2, pp. 1416–1423, 2019. [Online]. Available: https://doi.org/10.1021/acs.analchem.8b04088

[48] J. Pánek, I. Eidhammer, and R. Aasland, "A new method for identification of protein (sub)families in a set of proteins based on hydropathy distribution in proteins," *Proteins: Structure, Function, and Bioinformatics*, vol. 58, no. 4, pp. 923–934, 2005. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.20356

[49] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[50] A. Rives, J. Meier, T. Sercu, S. Goyal, Z. Lin, J. Liu, D. Guo, M. Ott, C. L. Zitnick, J. Ma, and R. Fergus, "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences," *PNAS*, 2019. [Online]. Available: https://www.biorxiv.org/content/10.1101/622803v4

[51] H. Stärk, C. Dallago, M. Heinzinger, and B. Rost, "Light attention predicts protein location from the language of life," *Bioinformatics Advances*, vol. 1, no. 1, 11 2021, vbab035. [Online]. Available: https://doi.org/10.1093/bioadv/vbab035

[52] K. Weißenow, M. Heinzinger, and B. Rost, "Protein language model embeddings for fast, accurate, alignment-free protein structure prediction," *bioRxiv*, 2021. [Online]. Available: https://www.biorxiv.org/content/early/2021/08/02/2021.07.31.454572

[53] J. Meier, R. Rao, R. Verkuil, J. Liu, T. Sercu, and A. Rives, "Language models enable zero-shot prediction of the effects of mutations on protein function," *bioRxiv*, 2021. [Online]. Available: https://www.biorxiv.org/content/early/2021/11/17/2021.07.09.450648

[54] C. Hsu, R. Verkuil, J. Liu, Z. Lin, B. Hie, T. Sercu, A. Lerer, and A. Rives, "Learning inverse folding from millions of predicted structures," *bioRxiv*, 2022. [Online]. Available: https://www.biorxiv.org/content/early/2022/09/06/2022.04.10.487779

[55] P. Väth, M. Münch, C. Raab, and F.-M. Schleif, "Proval: A framework for comparison of protein sequence embeddings," *Journal of Computational Mathematics and Data Science*, vol. 3, p. 100044, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2772415822000128

[56] M. Remmert, A. Biegert, A. Hauser, and J. Sding, "Hhblits: Lightning-fast iterative protein sequence searching by hmm-hmm alignment," *Nature Methods*, vol. 9, no. 2, p. 173–175, 2011.

[57] J. A. Capra and M. Singh, "Predicting functionally important residues from sequence conservation," *Bioinformatics*, vol. 23, no. 15, pp. 1875–1882, 05 2007. [Online]. Available: https://doi.org/10.1093/bioinformatics/btm270

[58] W. R. Taylor, "The classification of amino acid conservation," *Journal of Theoretical Biology*, vol. 119, no. 2, pp. 205–218, 1986. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022519386800753

[59] J. Yang, A. Roy, and Y. Zhang, "Protein–ligand binding site recognition using complementary binding-specific substructure comparison and sequence profile alignment," *Bioinformatics*, vol. 29, no. 20, pp. 2588–2595, 08 2013. [Online]. Available: https://doi.org/10.1093/bioinformatics/btt447

[60] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of Big Data*, vol. 6, no. 1, 2019.

[61] S. Nadendla, R. Jackson, J. Munro, F. Quaglia, B. Mészáros, D. Olley, E. Hobbs, S. Goralski, M. Chibucos, C. Mungall, S. C. E. Tosatto, I. Erill, and M. Giglio, "ECO: the Evidence and Conclusion Ontology, an update for 2022," *Nucleic Acids Research*, vol. 50, no. D1, pp. D1515–D1521, 11 2021. [Online]. Available: https://doi.org/10.1093/nar/gkab1025

[62] J. Hastings, G. Owen, A. Dekker, M. Ennis, N. Kale, V. Muthukrishnan, S. Turner, N. Swainston, P. Mendes, and C. Steinbeck, "ChEBI in 2016: Improved services and an expanding collection of metabolites," *Nucleic Acids Research*, vol. 44, no. D1, pp. D1214–D1219, 10 2015. [Online]. Available: https://doi.org/10.1093/nar/gkv1031

[63] B. Jagannathan and J. Golbeck, "Fx, fa, and fb iron–sulfur clusters in type i photosynthetic reaction centers," in *Encyclopedia of Biological Chemistry (Second Edition)*, second edition ed., W. J. Lennarz and M. D. Lane, Eds. Waltham: Academic Press, 2013, pp. 335–342. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780123786302001845

[64] Soedinglab, "Mmseqs2: ultra fast and sensitive sequence search and clustering suite," Apr 2023. [Online]. Available: https://github.com/soedinglab/MMseqs2

[65] D. Ofer, N. Brandes, and M. Linial, "The language of proteins: Nlp, machine learning protein sequences," *Computational and Structural Biotechnology Journal*, vol. 19, pp. 1750–1758, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2001037021000945

[66] D. Sgarbossa, U. Lupo, and A.-F. Bitbol, "Generative power of a protein language model trained on multiple sequence alignments," 2022.

[67] A. Elnaggar, H. Essam, W. Salah-Eldin, W. Moustafa, M. Elkerdawy, C. Rochereau, and B. Rost, "Ankh: Optimized protein language model unlocks general-purpose modelling," 2023.

[68] B. E. Suzek, Y. Wang, H. Huang, P. B. McGarvey, C. H. Wu, and the UniProt Consortium, "UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches," *Bioinformatics*, vol. 31, no. 6, pp. 926–932, 11 2014. [Online]. Available: https://doi.org/10.1093/bioinformatics/btu739

[69] L. Yang, W. He, Y. Yun, Y. Gao, Z. Zhu, M. Teng, Z. Liang, and L. Niu, "Defining a global map of functional group-based 3d ligand-binding motifs," *Genomics, Proteomics Bioinformatics*, vol. 20, no. 4, pp. 765–779, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1672022922000195

[70] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[71] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016.

[72] L. Liu, J. Liu, and J. Han, "Multi-head or single-head? an empirical comparison for transformer training," 2021.

[73] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2019.

[74] ——, "Sgdr: Stochastic gradient descent with warm restarts," 2017.

[75] I. Schlag, P. Smolensky, R. Fernandez, N. Jojic, J. Schmidhuber, and J. Gao, "Enhancing the transformer with explicit relational encoding for math problem solving," Nov. 2020, arXiv:1910.06611 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1910.06611

Appendix A

Other Comparison Results

Table A.1: Transformer encoder model (TFE) vs. Transformer encoder model trained on longer epoch (TFE-L), evaluated on the test set. Precision, recall, specificity, and MCC calculated using a threshold of 0.5. All metrics are rounded to three decimal places. Bold fonts indicate the best results.

Metal Type	Predictor	Precision	Recall	Specificity	MCC	AUROC	AUPRC
Zn(2+)	TFE	42.7%	**65.3%**	100%	**0.528**	**0.971**	**0.529**
	TFE-L	**45.1%**	58.8%	100%	0.515	0.969	0.511
Mg(2+)	TFE	33.0%	**57.5%**	100%	0.435	**0.945**	**0.451**
	TFE-L	**44.7%**	45.0%	100%	**0.449**	0.923	0.399
[4Fe-4S] cluster	TFE	40.5%	**90.9%**	100%	0.607	1.000	**0.652**
	TFE-L	**50.5%**	81.8%	100%	**0.643**	1.000	0.570
Ca(2+)	TFE	32.0%	**52.3%**	100%	**0.409**	**0.969**	**0.340**
	TFE-L	**41.7%**	35.2%	100%	0.383	0.946	0.272
Mn(2+)	TFE	**12.7%**	**39.4%**	100%	**0.224**	**0.971**	**0.080**
	TFE-L	6.3%	11.3%	100%	0.085	0.959	0.039
a divalent metal cation	TFE	39.3%	**63.8%**	100%	0.501	**0.960**	**0.506**
	TFE-L	**47.2%**	53.1%	100%	0.501	0.959	0.495
Fe cation	TFE	**35.5%**	**38.6%**	100%	**0.370**	**0.956**	**0.226**
	TFE-L	20.0%	28.1%	100%	0.237	0.947	0.124
[2Fe-2S] cluster	TFE	**84.8%**	96.6%	100%	**0.905**	1.000	0.966
	TFE-L	65.1%	96.6%	100%	0.793	1.000	0.966
Cu cation	TFE	72.3%	**87.0%**	100%	0.793	0.959	**0.846**
	TFE-L	**81.8%**	83.3%	100%	**0.826**	**0.996**	0.843
K(+)	TFE	0.0%	0.0%	100%	0.000	0.445	0.000
	TFE-L	0.0%	0.0%	100%	0.000	**0.578**	0.000
Ni(2+)	TFE	0.0%	0.0%	100%	0.000	0.999	0.002
	TFE-L	0.0%	0.0%	100%	0.000	**1.000**	**0.003**
Na(+)	TFE	0.0%	0.0%	100%	0.000	0.831	0.000
	TFE-L	0.0%	0.0%	100%	0.000	**0.840**	0.000
Fe(3+)	TFE	0.0%	0.0%	100%	0.000	0.753	**0.001**
	TFE-L	0.0%	0.0%	100%	0.000	**0.782**	0.000
iron-sulfur cluster	TFE	52.7%	**92.5%**	100%	0.698	1.000	**0.831**
	TFE-L	**58.7%**	91.6%	100%	**0.733**	1.000	0.695
Cu(2+)	TFE	**16.7%**	**25.0%**	100%	**0.204**	1.000	0.235
	TFE-L	0.0%	0.0%	100%	0.000	1.000	**0.505**
Fe(2+)	TFE	0.0%	0.0%	100%	0.000	0.999	0.001
	TFE-L	0.0%	0.0%	100%	0.000	0.999	0.001
Co(2+)	TFE	0.0%	0.0%	100%	0.000	**0.886**	**0.003**
	TFE-L	0.0%	0.0%	100%	0.000	0.881	0.001
a metal cation	TFE	41.9%	**64.7%**	100%	**0.520**	0.935	**0.520**
	TFE-L	**49.3%**	54.5%	100%	0.518	**0.938**	0.506
Mean	TFE	28.0%	**43.0%**	100%	**0.344**	0.921	**0.344**
	TFE-L	**28.4%**	35.5%	100%	0.316	**0.929**	0.329

Table A.2: Transformer encoder model trained on Ankh embeddings (TFE) vs Transformer encoder model trained on ProtT5 embeddings (TFE-T5), evaluated on the test set. Precision, recall, specificity, and MCC calculated using a threshold of 0.5. All metrics are rounded to three decimal places. Bold fonts indicate the best results.

Ligand Type	Predictor	Precision	Recall	Specificity	MCC	AUROC	AUPRC
Zn(2+)	TFE	**42.7**%	65.3%	100%	**0.528**	0.971	**0.529**
	TFE-T5	35.6%	**69.9**%	100%	0.498	**0.978**	0.477
Mg(2+)	TFE	**33.0**%	**57.5**%	100%	**0.435**	**0.945**	**0.451**
	TFE-T5	22.6%	51.2%	100%	0.340	0.918	0.317
[4Fe-4S] cluster	TFE	**40.5**%	90.9%	100%	**0.607**	1.000	**0.652**
	TFE-T5	25.1%	**93.9**%	100%	0.486	1.000	0.611
Ca(2+)	TFE	**32.0**%	**52.3**%	100%	**0.409**	**0.969**	**0.340**
	TFE-T5	26.7%	50.8%	100%	0.368	0.949	0.317
Mn(2+)	TFE	**12.7**%	**39.4**%	100%	**0.224**	**0.971**	**0.080**
	TFE-T5	7.1%	35.2%	100%	0.158	0.930	0.032
a divalent metal cation	TFE	**39.3**%	63.8%	100%	**0.501**	**0.960**	**0.506**
	TFE-T5	31.4%	63.8%	100%	0.447	0.949	0.434
Fe cation	TFE	**35.5**%	**38.6**%	100%	**0.370**	0.956	**0.226**
	TFE-T5	20.2%	36.8%	100%	0.273	**0.982**	0.180
[2Fe-2S] cluster	TFE	**84.8**%	**96.6**%	100%	**0.905**	**1.000**	**0.966**
	TFE-T5	62.8%	93.1%	100%	0.765	0.993	0.914
Cu cation	TFE	72.3%	**87.0**%	100%	0.793	0.959	0.846
	TFE-T5	**76.3**%	83.3%	100%	**0.797**	**0.999**	**0.849**
K(+)	TFE	0.0%	0.0%	100%	0.000	0.445	0.000
	TFE-T5	0.0%	0.0%	100%	0.000	**0.760**	0.000
Ni(2+)	TFE	0.0%	0.0%	100%	0.000	0.999	**0.002**
	TFE-T5	0.0%	0.0%	100%	0.000	0.999	0.001
Na(+)	TFE	0.0%	0.0%	100%	0.000	0.831	0.000
	TFE-T5	0.0%	0.0%	100%	0.000	**0.977**	0.000
Fe(3+)	TFE	0.0%	0.0%	100%	0.000	0.753	0.001
	TFE-T5	0.0%	0.0%	100%	0.000	**0.808**	**0.012**
iron-sulfur cluster	TFE	**52.7**%	92.5%	100%	**0.698**	**1.000**	0.831
	TFE-T5	34.4%	**93.5**%	100%	0.567	0.998	**0.869**
Cu(2+)	TFE	**16.7**%	**25.0**%	100%	**0.204**	1.000	**0.235**
	TFE-T5	0.0%	0.0%	100%	0.000	1.000	0.220
Fe(2+)	TFE	0.0%	0.0%	100%	0.000	0.999	0.001
	TFE-T5	0.0%	0.0%	100%	0.000	**1.000**	**0.002**
Co(2+)	TFE	0.0%	0.0%	100%	0.000	0.886	0.003
	TFE-T5	0.0%	0.0%	100%	0.000	**0.946**	**0.004**
a metal cation	TFE	**41.9**%	**64.7**%	100%	**0.520**	**0.935**	**0.520**
	TFE-T5	33.8%	64.0%	100%	0.464	0.925	0.454
Mean	TFE	**28.0**%	**43.0**%	100%	**0.344**	0.921	**0.344**
	TFE-T5	20.9%	40.8%	100%	0.287	**0.951**	0.316

Appendix B

Development Code and Datasets

The full development code and datasets are accessible at https://github.com/zhuzihan728/COMP0138-Metal-Binding-Site-Prediction.git.
The development tools utilized are summarized in Tab B.1.

Table B.1: Main tools utilized in the project development.

Programming language	python
Protein clustering & similarity search	mmseqs
Package management	conda
Program execution	Jupyter notebook, colab
Python libraries	
Data preprocessing	pandas, biopython
Feature representation	transformers, Ankh
Model training	pytorch, pytorch lightning, tensorboard
Validation	scikit-learn
Data visualization	matplotlib, seaborn
File IO	h5py
Data serialization	json, yaml
Mathematical operations	numpy, scipy