Accurate Protein Function Prediction with Graph Transformer-Based Function Localization

by

Shania Mitra

Submitted to the Center for Computational Science and Engineering
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN COMPUTATIONAL SCIENCE AND ENGINEERING

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2025

© 2025 Shania Mitra. All rights reserved.

The author hereby grants to MIT a nonexclusive, worldwide, irrevocable, royalty-free license to exercise any and all rights under copyright, including to reproduce, preserve, distribute and publicly display copies of the thesis, or release the thesis under an open-access license.

Authored by: Shania Mitra
Center for Computational Science and Engineering
May 14, 2025

Certified by: Manolis Kellis
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Certified by: Connor Coley
Associate Professor of Chemical Engineering,
Electrical Engineering and Computer Science
Thesis Reader

Accepted by: Nicolas Hadjiconstantinou
Professor, Mechanical Engineering
Co-Director, Center for Computational Science and Engineering

This page was intentionally left blank

THESIS COMMITTEE

Thesis Supervisor

Manolis Kellis
Professor of Computer Science
Department of Electrical Engineering and Computer Science

Thesis Readers

Connor W. Coley
Associate Professor of Chemical Engineering,
Electrical Engineering and Computer Science

This page was intentionally left blank

Accurate Protein Function Prediction with Graph Transformer-Based Function Localization

by

Shania Mitra

Submitted to the Center for Computational Science and Engineering
on May 14, 2025 in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN COMPUTATIONAL SCIENCE AND ENGINEERING

ABSTRACT

Protein function prediction is a fundamental challenge in biology, crucial for understanding biological processes, disease mechanisms, and accelerating drug discovery. While computational methods leveraging sequence or structural information have advanced, accurately translating protein structure to function and pinpointing the specific residues responsible remain significant hurdles. Many existing deep learning approaches fall short, often relying on post-hoc analyses that lack specificity or fail to directly integrate functional site identification into the prediction process.

In this study, we introduce the Protein Region Proposal Network (ProteinRPN), a novel graph-based deep learning framework designed to address these limitations. ProteinRPN is the first model to integrate the proactive identification of functional regions within the Gene Ontology term prediction pipeline. The core of the model is a Region Proposal Network module that processes protein structure graphs (residues as nodes, contacts as edges) to identify potential functional regions, termed anchors. These anchors are subsequently refined using a multi-stage process involving a novel differentiable node drop pooling layer that incorporates domain knowledge. A functional attention layer further enhances the representations of predicted functional nodes, and a Graph Multiset Transformer aggregates this localized information into a comprehensive graph-level embedding for final prediction. The model is optimized using a combination of cross-entropy classification loss, supervised and self-supervised contrastive learning losses (SupCon and InfoNCE) for robust representation learning.

Evaluated on standard benchmarks derived from the DeepFRI/HEAL datasets, ProteinRPN demonstrates state-of-the-art performance, consistently outperforming existing sequence-based and structure-based methods across all three Gene Ontology domains (Molecular

Function, Biological Process, Cellular Component) based on standard CAFA metrics (F_{max}, AUPR, S_{min}). Notably, ProteinRPN achieves significant improvements over strong baselines like HEAL, with AUPR (Area under Precision Recall curve) gains of approximately 15.4% (BP), 8.5% (CC), and 1.3% (MF). Furthermore, ablation studies validate the contribution of each key component, particularly the region proposal mechanism. Qualitative analysis confirms the model's ability to accurately localize known functional residues within protein structures, offering enhanced interpretability.

By directly modeling and identifying functionally relevant structural regions, ProteinRPN presents a robust, interpretable, and high-performing approach to structure-based protein function prediction. This work contributes a novel framework that bridges the gap between structural information and functional annotation, offering potential for deeper biological insights and advancing computational tools for understanding the proteome.

Thesis supervisor: Manolis Kellis

Title: Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Thesis supervisor: Connor Coley

Title: Associate Professor of Chemical Engineering,
Electrical Engineering and Computer Science
Thesis Reader

Acknowledgments

I am deeply grateful to my advisor, Prof. Manolis Kellis, for his invaluable guidance and unwavering support throughout this research. I am also indebted to my collaborator, Lei Huang, for his constant mentorship and help with executing our collaborative work. I would like to thank my thesis reader, Prof. Connor Coley, for his insightful suggestions that significantly improved this thesis.

I am also thankful to my labmates, Giulianna Hashemi-Asasi and Robert Calef, for their helpful brainstorming and support.

Finally, heartfelt thanks to my parents and friends for their constant encouragement and belief in me during this journey.

This page was intentionally left blank

Contents

List of Figures

List of Tables

This page was intentionally left blank

Chapter 1

Introduction

Proteins are essential biomolecules constituting living organisms, playing a plethora of roles within biological systems - such as catalyzing chemical reactions, participating in signal transduction, and maintaining cellular structure. The exact number of proteins in nature is unknown, but current estimates range between millions to billions. An early study in 2007 concluded that there are around 5 million sequences in the earth's proteome [1]. Currently, there are 350 million unique protein coding sequences in UniProt [2] and over 3 billion in MGnify [3]. It is estimated that the human proteome consists of approximately 20,000 genes, leading to approximately 70,000 proteins [4] serving a variety of functions. For instance, proteins provide structural support for cells and tissues, forming structures like *actin* and *tubulin* in the cytoskeleton, and *collagen* and *elastin* in connective tissues. They also act as *enzymes*, which are biological catalysts that speed up chemical reactions in the body. They are also vital as transport molecules like *hemoglobin* that carries oxygen in the blood, or *transport proteins* that help move substances across cell membranes. Proteins, such as *antibodies*, are essential for the immune system, helping the body fight off infections. Proteins are needed for growth, repair, and maintenance of tissues, as they provide the building blocks for new cells. Further, proteins like *actin* and *myosin* are involved in muscle contraction, enabling movement [5].

Thus, determining the functions of proteins is essential in understanding and uncovering biological systems. They are also common targets for drug discovery since many diseases are facilitated by malfunctioning proteins, and thus, information on their function can facilitate the development of drugs targeting these proteins. Thus, they are central to biotechnological advancement [6].

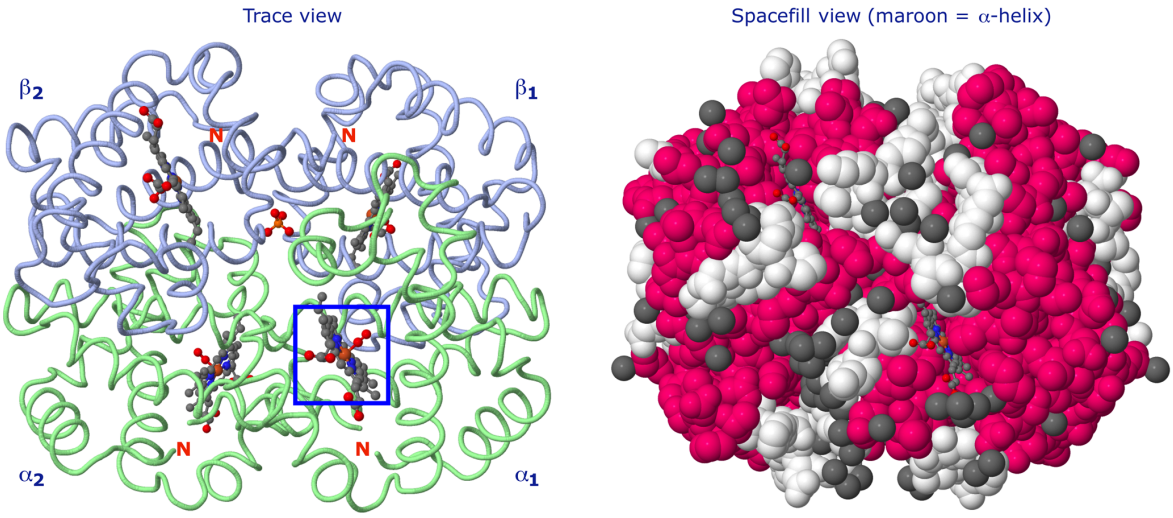

Figure 1.1: Sample 3D structure visualization: Hemoglobin. The trace view (left) displays the four subunit backbones and heme groups, while the spacefill view (right) emphasizes the protein's shape and surface accessibility.

1.1 The Challenge of Understanding Protein Function

Proteins are polymers of 20 common amino acids, typically ranging in length from 50 to 2000 residues [7]. They are usually expressed as a linear sequence of these 20 amino acids known as the primary structure of the protein, bonded together using peptide bonds. These chains then fold due to the formation of strong covalent disulfide bonds, forming local sheet or helix arrangements known as secondary structures. These local conformations come together in 3D space to form a coiled and highly complex tertiary structure of the protein [5]. These complex 3D structures then interact with one another through non-covalent forces and assemble into complexes that provide the machinery for proteins to function and life to take place [8].

Protein function prediction is essential for understanding both intracellular and extracellular biological processes, as well as the relationship between protein diversity and evolution. It also contributes to elucidating gene functions, mapping gene regulatory networks, and annotating uncharacterized proteins. Accurate predictions support drug discovery by identifying relevant protein targets, improving the design and effectiveness of therapeutics. In personalized medicine, insights into individual variation in protein function are critical for assessing disease risk and designing tailored treatment strategies [8].

As the field has progressed over the past several years, as a result of various genome and

transcriptome projects, we now have access to >250 million protein sequences. However, less than 1% (∼170k) protein sequences have experimentally determined function. This presents a large gap between known sequences and their functions known as the sequence-function gap [6]. While protein structure can be resolved using experimental methods like X-ray crystallography, protein function is typically determined through techniques such as biochemical assays and enzymatic analysis. However, these functional assays are often costly, time-consuming, and labor-intensive, limiting their applicability to only a small subset of proteins. This makes accurate computational prediction of protein function a critical and challenging task.

To standardize protein function across studies, the Gene Ontology (GO) framework is commonly used. GO categorizes protein function into three main sub-ontologies: Biological Process (BP), which describes the broader biological goals a protein contributes to; Molecular Function (MF), which refers to the specific biochemical activity of the protein; and Cellular Component (CC), which indicates where within the cell the protein is active [8]. These are further explained in Chapter 2.

1.2 Overview of Current Protein Function Prediction Approaches

Since experimental techniques are expensive, time-consuming and labor-intensive [6, 8], computational methods have become ubiquitous due to their high accuracy, scalability and robust performance. Over the past decade, many such computational methods have been proposed to address the sequence function gap. These are broadly classified into sequence-based methods, 3D structure-based methods, Protein-Protein Interaction (PPI) network-based methods and hybrid information-based methods [8].

1.2.1 Sequence-based methods

Among computational approaches, sequence-based methods have been the most widely used for protein function prediction, primarily due to the universal availability of protein sequences. These methods aim to extract functionally relevant features directly from the amino acid sequence, often capturing subtle patterns that correlate with functional roles. While additional data types, such as 3D structure, can offer complementary information to enhance predictions,

sequence remains the primary modality used to predict function given its accessibility across virtually all known proteins [6, 8]. The accuracy of these methods, however, is limited.

1.2.2 Structure-based methods

Structure-based methods, on the other hand, leverage 3D structural information to predict protein function. These approaches often represent protein structures as contact maps, enabling the extraction of features that are closely tied to function. While protein sequence is frequently included as an additional input, structure-based methods offer distinct advantages for proteins whose function is heavily influenced by spatial conformation and structural context [8].

1.2.3 Protein–protein interaction-based methods

Protein–protein interaction (PPI) network-based methods leverage interaction data to improve protein function prediction. Since proteins rarely function in isolation, incorporating PPI information is particularly valuable for predicting Gene Ontology terms related to Biological Processes (BP), which often involve the coordinated activity of multiple proteins. These methods typically operate on genome-scale interaction networks, aggregating data from multiple sources to capture the functional organization of proteins. It has been shown that interacting proteins in PPI networks tend to share similar functions, making it feasible to infer a protein's function based on its interaction partners. As with structure-based methods, protein sequence is often used as an additional input to enrich the model. Some approaches further emphasize integrating heterogeneous information from various interaction types to improve functional predictions [8]. However, the reliability of PPI-based predictions can be affected by the quality of the interaction data. High-throughput experimental techniques used to generate PPI networks may introduce noise, potentially impacting the accuracy of the predicted functions [6].

1.2.4 Hybrid Information-based methods

Hybrid information-based methods integrate multiple sources of data—such as protein sequences, 3D structures, PPI networks, and InterPro features—to improve the accuracy and

robustness of protein function prediction. These integrative approaches aim to leverage the complementary strengths of each modality [8].

A detailed discussion of all the above methods is provided in Chapter 2.

1.3 Limitations and Opportunities

Even after significant advancements, many challenges in predicting protein function accurately remain. These include the absence of clear features correlated with certain functions, making it difficult to design effective algorithms; the existence of homologous proteins with minor sequence or structural differences that lead to distinct functions; the multifunctionality of many proteins; and the limited availability of functional assays, representative structures, or annotations for a large fraction of known proteins [9].

For many biochemical and biophysical functions, no well-defined sequence or structural motifs have been identified that consistently correlate with those functions. In some cases, this is due to the weak conservation of the physical requirements underlying the function. For instance, protein–protein interaction sites often involve relatively smooth and poorly conserved surface regions, making it harder to predict them using surface pocket detection or motif-based methods, unlike catalytic sites, which are typically more conserved and easier to localize. Additionally, functional activity is often localized to small, specific regions within a protein. For example, in several proteins, the ability to bind plasminogen has been traced primarily to the presence of a lysine residue at the protein terminus [9].

Another significant challenge in protein function prediction is the limitation of homology-based methods [9]. Even proteins with high sequence or structural similarity can exhibit different functions. Small variations in or near the active site can alter a protein's activity, resulting in a different substrate specificity or catalytic mechanism. For instance, members of the enolase superfamily share a common evolutionary origin, a conserved $(\beta/\alpha)_7\beta$-barrel (TIM-barrel) fold, and similar active site architectures. Yet, they catalyze distinct biochemical reactions. This superfamily includes:

- Enolases, which convert 2-phosphoglycerate to phosphoenolpyruvate in glycolysis
- Muconate lactonizing enzymes, which break down aromatic compounds into citric acid cycle intermediates

- Mandelate racemases, which interconvert (S)- and (R)-mandelate enantiomers
- 3-methylaspartate ammonia lyases, which are responsible for the breakdown of L-threo-3-methylaspartate into mesaconate and ammonia
- As well as several other functionally diverse enzymes [10, 11]

Similarly, the aminotransferase family includes paralogs that, despite sharing conserved active site residues, perform distinct functions. These examples highlight the limitations of relying solely on homology and highlight the need for more nuanced models that can capture subtle functional differences and understand key functional residues underlying certain function patterns [9].

Most existing protein function methods lack the ability to identify key functional residues. Those that do, often rely on post-hoc interpretability techniques, such as gradient-based Class Activation Maps (CAMs) [12, 13], to highlight which residues contribute most to a predicted function. While these visual explanations offer some insight, they are retrospective in nature and reflect only what the model has implicitly learned during training, without incorporating prior biological knowledge about known functional sites. As a result, these methods often highlight many scattered residues with low specificity, which dilutes interpretability and reduces performance by failing to concentrate on biologically meaningful regions.

1.4 ProteinRPN: A Graph Region Proposal Approach for Protein Function Prediction

To address the limitations of existing approaches, we propose ProteinRPN, a novel model for accurate protein function prediction that explicitly incorporates functional residue detection. By focusing on regions where groups of residues collaborate to carry out specific biological functions, ProteinRPN enhances interpretability and precision. Drawing inspiration from region proposal networks in computer vision [14], the model introduces a graph-based Region Proposal Network (RPN) to identify candidate functional subregions within protein graphs.

The model begins by proposing k-hop subgraphs (anchors) centered on each node, aimed at capturing regions likely to contain functional residues. These anchors are then refined using a node drop pooling mechanism to prune nodes in the large anchors to identify the sparsely present functional residues. Once the graph is enriched with key functional residue

information, to generate global protein representations, Graph Multiset Transformer (GMT) aggregates the node-level embeddings, preserving both localized functional signals and the overall graph topology. Additionally, contrastive learning is employed to ensure that proteins with similar functions are mapped to nearby representations, while dissimilar proteins remain well separated in the embedding space.

The region proposal module is first pretrained on a functionally annotated dataset – the PDBSite dataset [15], which provides functional residue annotations from the Protein Data Bank (PDB) [16], a widely used repository of experimentally validated protein structures. We then evaluate ProteinRPN on the same benchmark datasets used by prior work [12, 13] to ensure fair comparison with state-of-the-art models.

1.5 Contributions and Thesis Outline

The experimental results indicate significant improvements in predicting protein functions compared to state-of-the-art (SOTA) models. Remarkably, the proposed model achieves a $\sim$15% improvement in protein-centric AUPR on BP $\sim$9% improvement in Fmax on the BP and MF ontologies compared to SOTA models. We also visualize the predicted functional residues, demonstrating that our model can identify essential functional structures and regions, which are meaningful for biological analysis.

This thesis is organized as follows. Chapter 2 provides background and surveys related work in protein function prediction. Chapter 3 presents the proposed methodology, detailing the architectural components and design choices behind our model, ProteinRPN. Chapter 4 outlines the experimental setup, including the datasets, training procedures, and evaluation metrics. In Chapter 5, we analyze the model's performance and compare it against state-of-the-art baselines. Finally, Chapter 6 concludes the thesis by summarizing key findings and discussing potential future directions.

This page was intentionally left blank

Chapter 2

Background and Related Work

2.1 Fundamentals of Protein Structure and Function

Proteins are important macromolecules, specifically, biopolymers that are vital for human, plant, and animal life. The repeating units of these polymers, i.e., the monomers, are amino acids, of which 20 are commonly found in nature.

Amino acids: These are the fundamental repeating units of proteins. They have a common backbone structure consisting of a central carbon atom (the α-carbon). This carbon atom is covalently bonded to an amino group, a carboxyl group, a hydrogen atom, and a variable side chain ($-R$ group) which accounts for the many different types of amino acids, giving each its unique characteristics. The central α-carbon atom is a chiral center for all amino acids except glycine, which has a hydrogen atom for its variable $-R$ group. Since the α-carbon atom is asymmetric, both L and D stereoisomers exist for most amino acids. However, in natural proteins it is found that most amino acids exist in the L-configuration [18]. The nature of the -R group determines the chemical properties of each amino acid such as size, charge, polarity, hydrophobicity, and ability to bond with other amino acids. These in turn have strong influence on the ability of the protein to fold and function.

The 20 standard amino acids occurring naturally in proteins tend to be classified into groups based on the properties of the variable $-R$ group [18]:

- Nonpolar Aliphatic (Alanine, Valine, Leucine, Isoleucine)
- Aromatic (Phenylalanine, Tyrosine, Tryptophan)

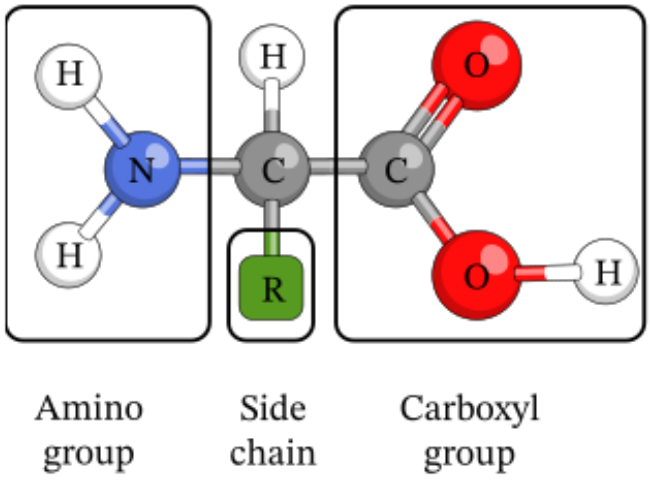

Figure 2.1: The core components of an amino acid monomer: the amino group, the carboxyl group, and the variable side chain (R-group) attached to the central alpha-carbon.

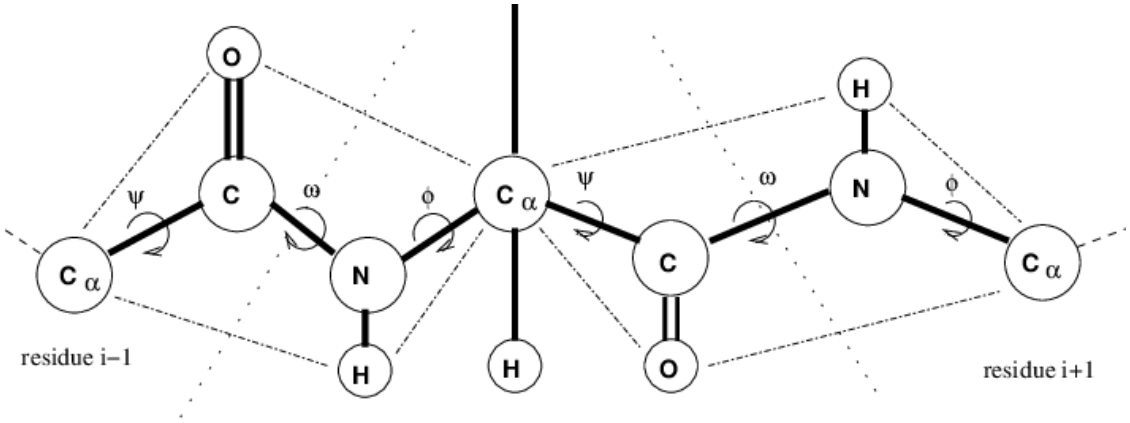

Figure 2.2: The polypeptide backbone illustrating the peptide plane and the main dihedral angles (phi ϕ, psi ψ, omega ω) that define backbone conformation.[17]

- Polar uncharged (Serine, Threonine, Cysteine, Glutamine, Asparagine)
- Positively charged or basic (Lysine, Arginine, Histidine)
- Negatively charged or acidic (Aspartate, Glutamate)

As is common in biological systems, there are always special cases: for example, proline consists of an aliphatic side chain which forms a rigid ring structure with the amino group, which introduces constraints in the backbone when many such units combine to form the protein polymer [19]. Another special case is that of cysteine, which contains a sulfhydryl (-SH) group known to form covalent disulfide bonds important for stabilizing tertiary and quaternary structures of proteins, discussed below.

Within proteins, the amino acid monomers link sequentially via peptide bonds. These are amide linkages formed through a dehydration reaction between the carboxyl group of one amino acid and the amino group of the next, creating a polypeptide chain with distinct N-terminus (free amino group) and C-terminus (free carboxyl group) ends [5, 7].

Proteins have a very strong biological significance and are responsible for life. A fundamental function of proteins is enzymatic catalysis. Enzymes are special proteins that are highly specialized and effective catalysts that catalyze specific biochemical reactions necessary for life. They exhibit extreme substrate specificity and regulatory control over metabolic pathways, and typically, each biochemical reaction is catalyzed by a specific enzyme. Other proteins serve important structural roles, providing shape and support to cells, tissues and organisms. For example, *collagen* in connective tissue, keratin in hair and nails, and cytoskeletal proteins like *actin* and *tubulin* that maintain cell shape and integrity. Aside from these vital functions, proteins are also essential for transport and storage, and routinely facilitate the movement of ions, small molecules and other proteins through membranes or through the bloodstream. For example, *hemoglobin* carries oxygen in the blood and *ferritin* stores iron.

Proteins are key in cell signaling and communication. Here, proteins act as hormones, neurotransmitters, and receptors embedded in cell membranes that detect external signals along with intracellular signals that relay messages within the cell. Examples include insulin which regulates glucose. Similarly, motor proteins are responsible for movement at both the cellular and organismal levels. For example, *myosin* interacting with *actin* for muscle contraction and *kinesin* and *dynein* for moving substances along microtubules within cells.

Proteins are also essential for immune defense in organisms. Antibodies or immunoglobulins are proteins designed to recognize and eliminate foreign organisms within the body, and

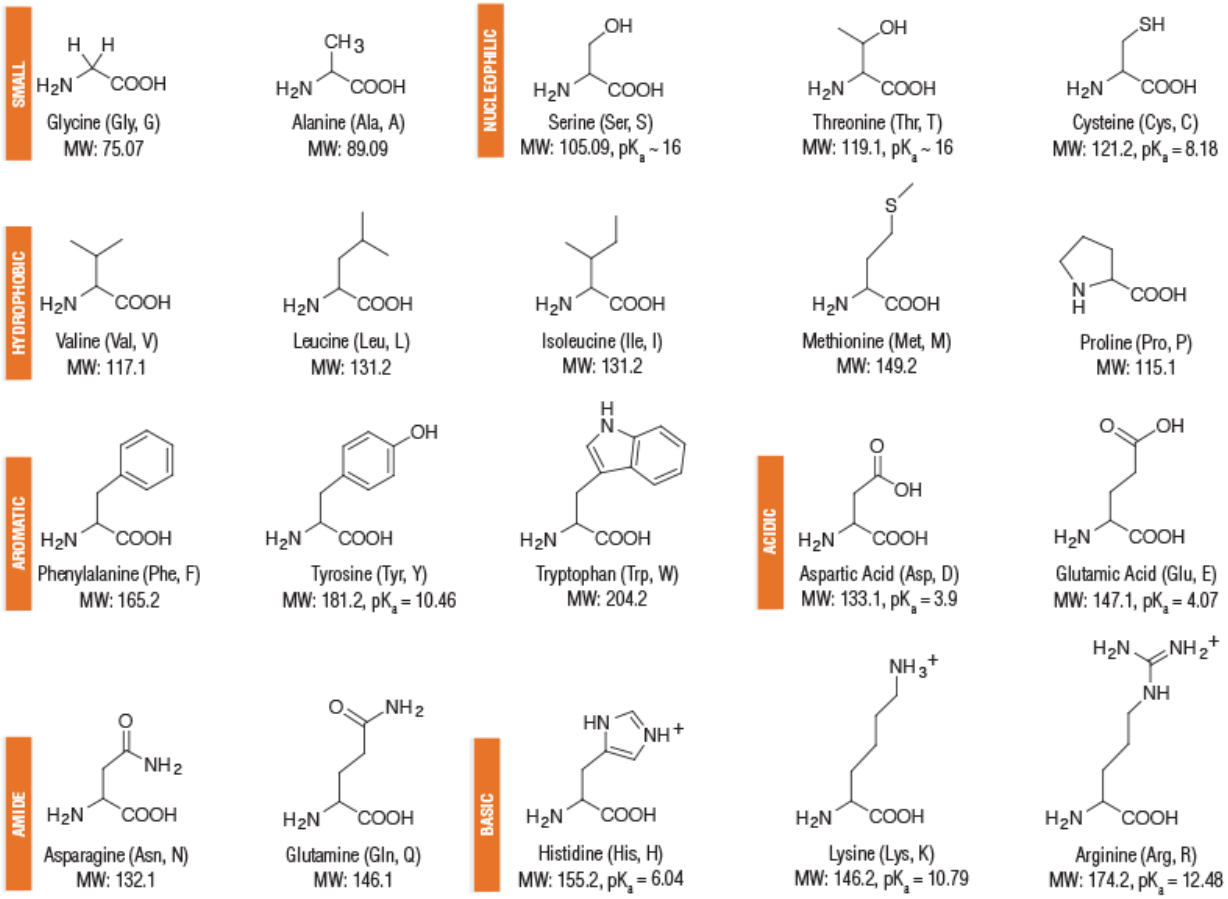

Figure 2.3: Chemical structures of the 20 standard amino acids commonly found in proteins. They are categorized based on the properties of their unique side chains ($-R$ groups) into groups such as small, hydrophobic, aromatic, acidic, basic, amide-containing, and nucleophilic. Molecular weight (MW) and relevant pK_a values are indicated.

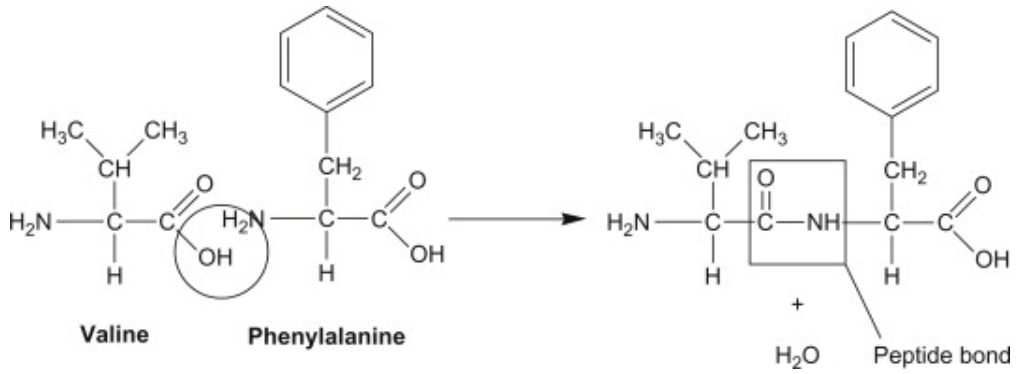

Figure 2.4: The condensation reaction forming a peptide bond. The α-carboxyl group of one amino acid (Valine) reacts with the α-amino group of another (Phenylalanine), eliminating a molecule of water (H2O) and creating a covalent amide linkage (the peptide bond).

cytokines complement them to regulate immune response. Thus, proteins play a wide variety of functions in organisms and are essential for their survival and functioning, necessitating accurate protein function annotation to understand biological processes more thoroughly, identifying drug targets and disease mechanisms.

2.1.1 Levels of Protein Structure

Protein function is very strongly linked to its three-dimensional structure [20]. This is because while folding into its 3D structure, the protein backbone exposes certain amino acid side chains into unique spatial configurations leading to those becoming active functional sites. These could be enzyme active sites, which bind to substrates and catalyze biochemical reactions, or binding pockets which facilitate other interactions such as those between DNA, RNA, other proteins or ligands. Hence, even minor alterations in the structure due to mutations or changes in the external environment could change the exposed active sites altering or inhibiting the functional activity of the protein. Thus, understanding the 3D structure of proteins is a prerequisite for strong functional understanding.

The 3D structure of proteins is conventionally described using a hierarchical framework that consists of four levels that build on top of each other. These are the primary, secondary, tertiary and quaternary structures.

Primary structure

This represents the most fundamental level comprising the linear sequence of amino acids that covalently bond together to form the polypeptide chain. The bonds, peptide bonds, are rigid planar amide linkages formed by the dehydration/condensation reaction between the α-carboxyl group of one amino acid and the α-amino group of the other amino acid as explained earlier. This creates the polypeptide backbone consisting of a repeating series of nitrogen, α-carbon, and carbonyl carbon atoms (N-C_α-C), from which the variable amino acid side chains (R-groups) project outward, interacting with one another and leading to higher levels of protein structure. As a convention, the primary structure is always written starting from the N-terminus towards the C-terminus (the end with the free α-carboxyl group), building directionality into the sequence. Thus, the primary structure gives us the recipe to understand the 3D structure of the proteins, as demonstrated in Anfinsen's experiments with ribonuclease [21–24].

Secondary structure

The secondary structure level is used to describe local spatial arrangements of the protein where adjacent protein residues of the sequence group together to form a segment and form certain helix or sheet like structures in 3D, independent of side chain conformations. These arrangements tend to be stabilized by hydrogen bonding between the carbonyl oxygen (C=O) atoms and the amide hydrogen (N-H) atoms within the backbone of the protein/polypeptide. The two most common such structures of spatial arrangements are the right-handed alpha-helix (α-helix) and the beta-pleated sheet (β-sheet) [23–25].

The α-helix is a tightly coiled, rod-like structure where the polypeptide backbone twists in a helix stabilized by hydrogen bonds between the C=O group of residue at the n^{th} position and the N-H group of residue at the $(n+4)^{th}$ position, with the R-groups projecting outwards from the helical axis, consisting of 3.6 residues per turn.

In contrast, the β-pleated sheet (β-sheet) is a flat sheet of residues formed by laterally packing two or more adjacent polypeptide segments known as β-strands. These strands lock together through hydrogen bonds between C=O and N-H groups on adjacent strands. The side chains of consecutive residues within a strand protrude alternately to opposite sides, roughly perpendicular to the plane of the sheet. These sheets can be of two types based on the relative orientation of the strands: anti-parallel, where adjacent strands run in opposite directions (N-terminus to C-terminus), and parallel, where adjacent strands run in the same direction. Anti-parallel β-sheets are known to have slightly greater stability since the hydrogen bonds are less strained [26, 27].

Segments within the polypeptide chains also form less regular structures known as loops or coils, or turns. These segments connect α-helices and β-strands and are usually found on the surface of the protein.

Tertiary structure

Tertiary structure refers to the overall 3D conformation adopted by a single protein or polypeptide chain, positioning each atom in the molecule. This spatial arrangement comes together due to the folding of the secondary structure elements and positioning of the amino acid side chains. This level describes the global folding pattern, detailing how the various α-helices, β-sheets, turns, and loops within the polypeptide are packed and arranged relative to one another in three-dimensional space. Unlike secondary structure which formed due to

hydrogen bonding in the backbone, tertiary structure is formed due to various interactions between the different side chain groups, often involving residues located far apart in the primary sequence, brought close together due to the folding process. Examples of these interactions include, hydrophobic interactions clustering nonpolar residues away from water, hydrogen bonds between polar side chains (and sometimes with backbone atoms), ionic bonds (salt bridges) between oppositely charged side chains, and weak van der Waals forces between closely packed atoms [27]. Additionally, covalent disulfide bonds are also important in deciding tertiary structure. These form between thiol groups (-SH) of specific cysteine residues that are close in the folded structure.

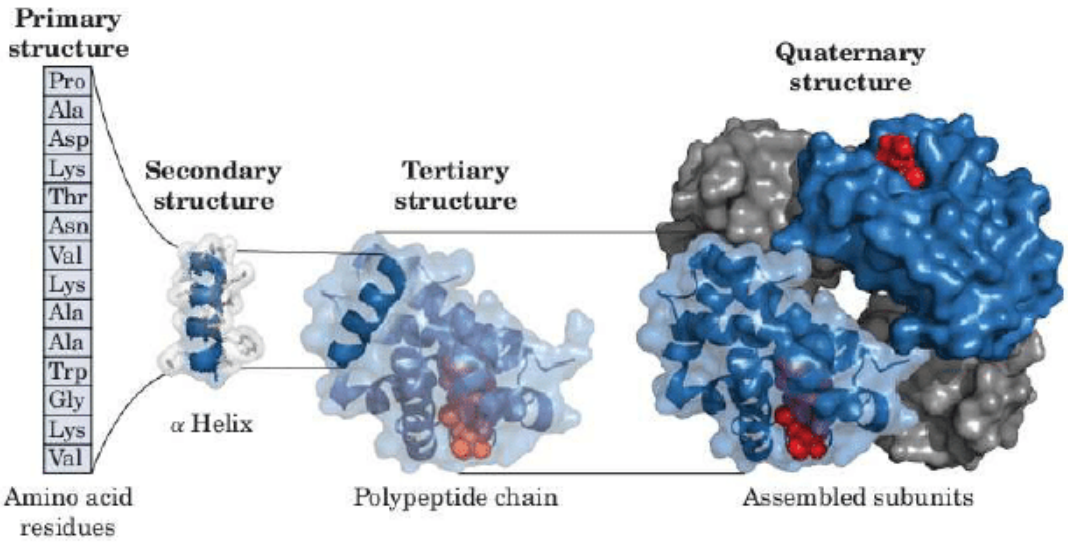

Figure 2.5: Illustration of protein structural organization. The linear sequence of amino acid residues (primary structure) folds into local secondary structures like the alpha-helix. These elements then fold into the overall three-dimensional shape of a polypeptide chain (tertiary structure). Finally, multiple polypeptide subunits can assemble to form a functional complex (quaternary structure)[28].

2.2 Frameworks for Protein Function Annotation

With the rapid advancement of genome sequencing technologies, we now have access to vast amounts of sequence data. Further, with the development of AlphaFold [29], we are now able to rapidly obtain the three-dimensional structures of these proteins to a high degree of accuracy. However, determining the function of these proteins still heavily relies on expensive and time-consuming experimental methods and is thus a major bottleneck in biological

research. Traditional experimental validation methods are often slow, resource-intensive, and difficult to apply on a large scale. This necessitates the development and application of accurate computational methods that are able to bridge this sequence-function gap.

2.2.1 Gene Ontology (GO): Structure and Usage

The Gene Ontology project [30, 31] is widely used to globally describe and classify the functions of gene products such as proteins. By providing a structured and controlled annotation system for function, it provides consistency to all methods classifying or describing protein function across species, allowing different methods and researchers to communicate in the "function" language easily. This is achieved using three distinct, orthogonal ontology domains that capture different aspects of function: Molecular Function (MF), Biological Process (BP), and Cellular Component (CC), explained further below.

- **Molecular Function (MF):** This ontology describes the particular biochemical activities performed by proteins at the molecular level. For example, 'ATP binding' or 'kinase activity'.
- **Biological Process (BP):** This represents broader biological objectives or pathways that the protein participates in through coordinated action of multiple molecular functions. Examples include 'DNA repair' or 'signal transduction'.
- **Cellular Component (CC):** This ontology defines the subcellular locations or structures where the protein is active. Example: the 'nucleus', 'cytoplasm', or 'mitochondrial inner membrane'.

Within each ontology, the GO terms are organized as a directed acyclic graph (DAG) where nodes represent terms from the ontology and edges signify hierarchical relationships between the terms such as "is a" or "part of". As we traverse the depth of the DAG, the terms become more and more specific, and annotation to a specific term automatically implies annotation to all its ancestor terms, referred to as the True Path Rule. More formally, it states that for a gene product to be annotated with a specific GO term, it must also be annotated with all of that term's parent terms (and subsequently all ancestor terms) up the hierarchy towards the root of the ontology.

For computational prediction, protein function annotation is often framed as a multi-label classification task using GO terms. However, this leads to a large number of labels since each

GO term forms a label and the hierarchical dependencies need to be handled. To mitigate this, each GO term is initially predicted independently and the model learns to predict the likelihood of the protein being associated with each specific term based on the input features. As a post-processing step, hierarchy enforcement is practiced in order to not violate the True Path Rule [32]. This is demonstrated in Algorithm 1: If a child term j is predicted positive (or has a score above a threshold), its parent term i must also be considered positive (its score might be adjusted upwards, e.g., $score(i) \geq score(j)$). This propagates positive predictions up the hierarchy. If a parent term i is predicted negative (or has a score below a threshold), all its descendant terms k must also be considered negative (their scores might be adjusted downwards). This propagates negative predictions down the hierarchy. This makes GO term prediction compatible with standard classification models, but requires an additional post-processing step in order to ensure compatibility with the real-world significance of GO terms.

Algorithm 1 Hierarchy Enforcement for True Path Rule

Require: Prediction scores $s(t)$ for each term t, threshold θ, hierarchy as a DAG with parent-child edges $i \rightarrow j$

for all terms j such that $s(j) \geq \theta$ **do**
  Propagate positive prediction upwards:
  for all ancestors i of j **do**
    $s(i) \leftarrow \max(s(i), s(j))$
  end for
end for
for all terms i such that $s(i) < \theta$ **do**
  Propagate negative prediction downwards:
  for all descendants k of i **do**
    $s(k) \leftarrow \min(s(k), s(i))$
  end for
end for
return Adjusted scores $s(t)$

2.2.2 Enzyme Commission (EC) Numbers: Classifying Enzymes

The enzyme commission (EC) numbers are an alternate labeling scheme applicable only to enzymes. It provides a hierarchical classification based specifically on chemical reactions catalyzed by enzymes. EC numbers are of the form X.X.X.X, consisting of four numerical components separated by periods. It specifies a four level hierarchy indicating the type of reaction catalyzed by an enzyme. The first component of the EC number defines the

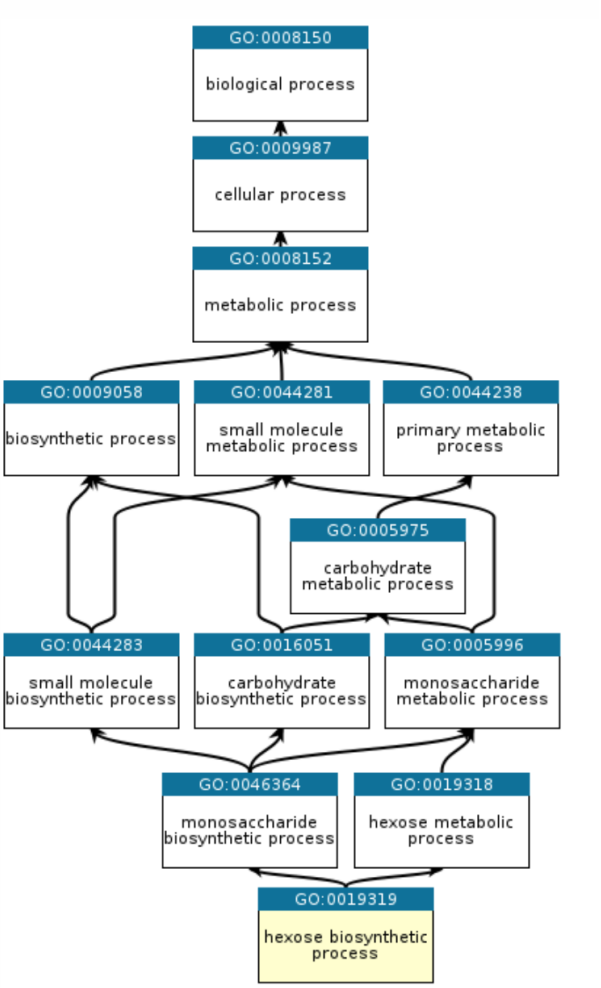

Figure 2.6: A subset of the Gene Ontology Biological Process hierarchy. This directed acyclic graph (DAG) shows relationships (indicated by arrows) between terms, starting from general concepts like 'metabolic process' (GO:0008152) and becoming progressively more specific down to 'hexose biosynthetic process' (GO:0019319).

main class of the enzyme. There are 6 main classes based on the general type of reaction: EC 1 (Oxidoreductases), EC 2 (Transferases), EC 3 (Hydrolases), EC 4 (Lyases), EC 5 (Isomerases), and EC 6 (Ligases). The second and third components specify the subclass and subsubclass which are related to the type of substrate or bond involved. The fourth digit is a serial number uniquely identifying the specific enzyme reaction. Example: EC 2.7.11.1 - protein-serine/threonine kinase.

Unlike the broader GO term classification, EC numbers limit themselves to enzymes and represent a stricter functional hierarchy. GO-based prediction provides a broader functional coverage across all protein-types and hence is the primary focus of this study.

2.2.3 Benchmarking Efforts: The CAFA Challenge

The Critical Assessment of protein Function Annotation (CAFA) [33] is a widely adopted experiment to benchmark leading protein function prediction methods in an objective manner. It is a time-based challenge where the organizers release a large set of protein sequences, including some whose functions are currently unknown. Participating teams submit function predictions as GO terms for these released sequences. Following this, wet-lab-based experimental evidence is collected for the target proteins leading to new or updated functional annotations. These are then collected in a benchmark set against which the submitted predictions are rigorously compared using standardized evaluation metrics. Today, these datasets are widely used to train, compare and benchmark leading protein function prediction models in the field. Common evaluation metrics used in CAFA, and adopted in this study, include protein-centric maximum F1-score (F_{max}), semantic distance (S_{min}), and Area Under the Precision-Recall curve (AUPR), which account for the hierarchical nature and imbalance of GO annotations [6, 34]. These are explained in detail in Chapter 4.

2.3 Machine Learning Concepts in Protein Function Prediction

In recent years, with the rise in machine learning-based approaches across biotechnology, machine learning, particularly deep learning, has proved to be beneficial in accurate and scalable annotation of function in proteins. These approaches offer powerful methods to learn complex patterns from large protein datasets and infer function computationally. Protein

Function Prediction is widely formulated as a supervised learning problem, i.e., a multi-label, multi-class classification problem, where the model learns to extract features from protein sequences or structures to output labels such as GO terms[33]. Architectures such as Convolutional Neural Networks (CNNs) [35], Graph Convolutional Networks (GCNs) [36], and Graph Attention Networks (GATs) [37] are very commonly used for the task, for example.

Attention mechanisms [38] are an important addition to newer deep learning models that allow the models to dynamically weigh the importance of different parts of the input data, such as specific amino acids in a sequence or key residues/regions in a structure when making predictions, often improving their efficacy.

Many modern deep learning approaches perform tasks that are a subset of the broader goal of representation learning, which aims to automatically learn meaningful embeddings or vector representations that capture the essential properties of proteins from raw data like sequences (e.g., using Protein Language Models like ESM-1, ESM-2 [39], ProtT5 [40], SAProt [41]) or structures. Techniques like contrastive learning, including Supervised Contrastive (SupCon) loss [42] and Noise-Contrastive Estimation (InfoNCE) [43], help train robust representations by explicitly encouraging embeddings of functionally similar proteins (or perturbed versions of the same protein) to be closer together while pushing dissimilar ones apart.

2.4 Related Work: Computational Methods in Protein Function Prediction

2.4.1 Protein Representation Learning

Deep learning models focused on protein representation learning aim to encode protein sequences and/or structures into vector representations that capture important information about proteins and can be used in downstream tasks requiring protein understanding, such as inputs for classification, regression or generation tasks.

Protein Representation learning models are primarily of three types: sequence representation learning, structure representation learning and multimodal representation learning [44].

Sequence representation learning models aim to build language models for proteins by training on diverse sets of sequences in order to generalize across diverse families. These could be

autoregressive language models (e.g., ProGen) [45], masked language models (e.g., ESM-1, ESM-2 [39], ProtTrans [40]) or Diffusion Language Models (e.g., DPLM) [46].

Structure Representation Learning models aim to encode protein structures by incorporating geometric and relational information about protein structures. Examples include EGNN [47], a graph neural network equivariant to Euclidean transformations respecting geometric symmetries, GVP [48] - learning from protein structures using geometric vector perceptrons, and GearNet [49] - which incorporated protein inductive bias through edges of the residue graph and incorporated geometric and relational information.

Finally, representations can also be learned through multimodal data. For example, ESM3 [50] reasons over sequence, structure, and function and generates proteins far from naturally occurring ones. ProST [51] is another model that learns protein representations by training on sequences and biomedical texts, enabling zero-shot protein classification and functional retrieval without annotations.

2.4.2 Protein Structure Prediction

Protein structure prediction or the protein folding problem is a fundamental challenge in biology aiming to accurately determine the 3D structure of a protein from its sequence by predicting the coordinates of every atom in the protein molecule. The central dogma of biology is that sequence determines structure, which in turn determines function [20]. Thus, as discussed above, knowing the structure of the protein is critical since it strongly influences the function and mechanism of action of the protein.

Experimental methods such as X-ray crystallography and cryo-EM can determine structures with high resolution, however, they are slow, expensive, and not applicable to all proteins, making it difficult to scale them to the millions of sequenced proteins [6].

Thus, computational methods became the need of the hour and several methods based on homology modeling (using templates of similar proteins), threading (fitting sequence to known folds), and ab initio methods (physics-based simulation) were proposed. Recent breakthroughs, however, have been based on deep learning. AlphaFold2 [29] marked one such groundbreaking advancement, achieving near-experimental accuracy for many single-chain protein predictions by leveraging attention mechanisms and co-evolutionary information from Multiple Sequence Alignments (MSAs). Alongside, RoseTTAfold [52] also emerged achieving high accuracy by simultaneously processing 1D sequence information, 2D distance/orientation

maps, and 3D coordinates.

Newer models such as OmegaFold [53] and ESM [39] evolved to get rid of the MSA requirement and instead use protein language models with geometry-focused transformers to predict structures directly from single sequences, which is beneficial for proteins with limited evolutionary data. The availability of highly accurate structure predictions from tools like these significantly benefits downstream tasks, including structure-based protein function prediction, providing essential 3D context even when experimental structures are unavailable.

2.4.3 Protein Function Prediction Methods

As discussed in Chapter 1, protein function prediction methods are of 4 types: sequence-based, structure-based, protein–protein interaction (PPI) network-based and hybrid, i.e., integrating two or more modalities [6, 8].

Sequence-Based Approaches

Traditional function prediction models used sequence-based methods [6, 54] and heavily relied on the principle of homology transfer, inferring function by identifying sequence similarity to already annotated proteins using tools like BLAST or DIAMOND. These methods were effective for proteins of close homologs, but had much lower efficacy at lower sequence identities (<30-40%). Further, these methods were unable to handle cases where proteins with similar sequences had different functions or vice versa, making them ungeneralizable. More recent sequence-based approaches have made use of machine learning and deep learning along with different sequence encodings (e.g., one-hot, k-mers, physicochemical properties) or learned embeddings from architectures like CNNs, RNNs, or Protein Language Models (PLMs) as input to classifiers. Examples include DeepGO [55] and DeepGOPlus [56] using CNNs [6], HiFun [57] combining BLOSUM and Word2Vec-based embeddings in CNN and BiLSTM architecture, and TALE [58] using transformers.

Structure-Based Approaches

Structure-based methods are based on the principle that protein 3D structure is more conserved than sequence and is a stronger and more direct determinant of molecular function

[54]. These methods analyze experimental or accurately predicted structures (e.g., from AlphaFold2 [29]) using techniques like structural alignment, motif searching, or geometric deep learning on graph or point cloud representations of the structure [6]. Graph Convolutional Networks (GCNs) and Graph Attention Networks (GATs), are commonly used to process protein structures represented as graphs where residues are nodes and the contact map defines edges [8].Examples include DeepFRI [12] using GCNs, GAT-GO using GATs and predicted contacts, HEAL [34] using message passing and graph transformers, Struct2GO [59] using GCNs with attention pooling [6].

Protein-Protein Interaction (PPI) Network-Based Approaches

Protein-Protein Interaction (PPI) network-based methods predict function based on the assumption that interacting proteins often share functions or participate in common biological pathways [60]. These methods utilize network topology, analyzing properties like direct neighbors, shared neighbors, network motifs, or applying graph algorithms like label propagation or network embedding. Examples include GeneMANIA [61], integrating multiple networks with label propagation, deepNF [62] using network diffusion/autoencoders and NetQuilt [63] using IsoRank for multi-species network alignment [6].

2.5 Current Gaps and Challenges in Protein Function Prediction

Even as computational methods in protein function prediction get more accurate and advanced, some key challenges remain, as highlighted in Jeffery et al. [9].

- **Persistent annotation gap:** Even as function detection methodologies get more advanced, a large fraction of protein-coding genes lack known functions. For example, greater than 30% for some genomes, while experimental validation remains slow and expensive.
- **Incompleteness and Bias in GO data:** GO term annotations are inherently incomplete, reflecting the Open World Assumption. This means that the absence of annotation does not mean the absence of function. It is possible that certain functions for a protein have not yet been discovered and only those that are known

have been annotated, rendering the annotations incomplete. Further, it is very difficult to experimentally guarantee that a protein does not have a certain function, thus, it is difficult to validate negative predictions and makes benchmarking complicated.

- **Ambiguity of specifying function:** The function of a protein is often context-dependent and subject to the environment of the protein, and many proteins, known as pleiotropic or "moonlighting" proteins possess multiple distinct functions, which is difficult to express using GO terms. Further, certain GO terms are associated with very few proteins making it difficult to generalize these functions to newer proteins.

- **Identifying specific functional determinants:** It is often very difficult to correlate sequence/structure with function even though that is the strongest known indicator. Small differences in sequence or structure can sometimes lead to vastly different functions (e.g., enzymes with similar folds but different specificities; high sequence similarity but different roles like Gal1/Gal3). Thus, it is extremely important to determine constellations of amino acids that play an active part in the protein's function. Any mutations in the protein that lead to a difference in the active site can heavily impair or modify the protein's function. To the best of our knowledge, ProteinRPN is the first model to proactively identify candidate active site regions using a graph proposal network for function determination.

In the next chapter we introduce our model ProteinRPN and discuss its architecture in detail, outlining the working of the model in identifying potential functional regions via region proposals, refining these proposals and finally generating GO term predictions.

Chapter 3

ProteinRPN Model Architecture and Methodology

In this chapter, we introduce ProteinRPN, a novel model for protein function prediction designed with biological insight. The core premise behind ProteinRPN is to focus on biologically meaningful regions of the protein, those likely to be functionally relevant, and use these regions to improve downstream prediction of Gene Ontology (GO) and Enzyme Commission (EC) terms.

We begin by modeling proteins as graphs, where nodes correspond to individual amino acid residues and edges are defined by the contact map, indicating spatial proximity (typically within 10 Å) between residues in the protein's three-dimensional structure.

As illustrated in Figure 3.1, ProteinRPN operates on these protein graphs and is composed of three key components:

- **Region Proposal Network (RPN):** This module identifies subgraphs (anchor regions) likely to contain functionally important residues. It systematically examines k-hop neighborhoods around each residue to propose candidate functional regions.
- **Function Attention Layer:** Following the identification of candidate functional residues, the Functional Attention Layer refines their representations by emphasizing interactions critical to protein function. It achieves this by adjusting edge attention weights based on connectivity to these predicted functional nodes, selectively amplifying their influence during feature aggregation.

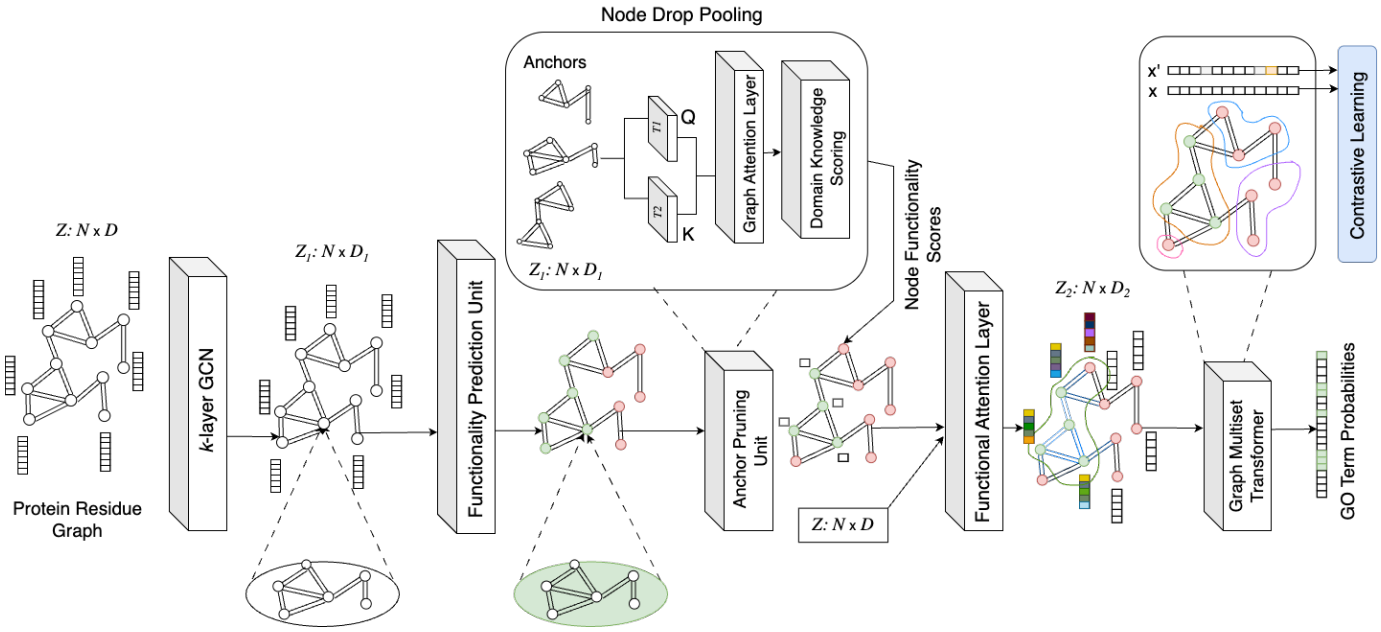

Figure 3.1: Overview of the ProteinRPN architecture, illustrating the main components: Region Proposal Network, Function Attention, and Function Prediction Module.

- **Function Prediction Module:** The refined residue representations are then aggregated using a Graph Multiset Transformer (GMT) pooling layer, followed by a multi-layer perceptron (MLP) readout to produce final predictions for GO and EC terms.

To train the model effectively, we use a combined loss function that includes both a Supervised Contrastive (SupCon) loss, which clusters proteins with similar functions, and a self-supervised InfoNCE loss, which enhances the quality of the learned representations by contrasting positive and negative protein pairs.

3.1 Motivation and Architectural Overview

Our architectural design is motivated by an analysis of 603 protein structures from the PDBSite dataset [15], which reveals that functional residues often form spatially clustered regions in the three-dimensional structure, even when they are not sequentially adjacent. In these proteins, each consisting of hundreds of residues, the number of functionally annotated residues ranges from 1 to 30.

These observations highlight several important considerations for protein function prediction.

First, they highlight the need to operate at the subgraph-level, rather than focusing on individual residues, since functional activity typically arises from localized clusters of residues within a protein's 3D environment. Second, the sparsity of functional residues relative to the total number of residues in the protein necessitates aggressive pruning to isolate meaningful regions. This calls for a multi-stage refinement pipeline that can progressively filter out irrelevant nodes while preserving biologically relevant clusters.

3.2 Input Representation and Protein Graph Construction

Proteins are represented as graphs $G(V, E)$, where the nodes V correspond to amino acid residues and the edges E denote spatial proximity between residues in three-dimensional space. The adjacency matrix $A \in \mathbb{R}^{N \times N}$ for a protein with N residues is constructed from the contact map: an undirected edge is added between two residues if the distance between their C_α atoms is less than 10 Å. In this work, we use $G(V, E)$ and $G(Z, A)$ interchangeably, where $Z \in \mathbb{R}^{|V| \times D}$ represents the node features and $A \in \mathbb{R}^{|V| \times |V|}$ is the adjacency matrix; D is the dimensionality of the residue embeddings.

The goal of ProteinRPN is to predict a probability vector $\hat{y}_i^{(j)} \in \mathbb{R}^{l_j}$, where l_j is the number of Gene Ontology (GO) terms for subontology $j \in \{BP, CC, MF\}$. The output vector $\hat{y}_i^{(j)}$ contains the predicted probabilities for each GO term in subontology j, capturing the likelihood that a protein is annotated with multiple terms across the three GO domains.

3.2.1 Residue Features

For each protein graph, node features for the N residues are constructed through a two-step process. First, each residue is encoded using ESM-1b embeddings to capture secondary and tertiary structure $Z_E \in \mathbb{R}^{N \times D_E}$, which provide context-aware sequence representations derived from a protein language model [64]. In parallel, residues are label-encoded based on their amino acid identities and embedded into vectors $Z_R \in \mathbb{R}^{N \times D_R}$. Both sets of embeddings are projected into a shared D-dimensional space and combined as follows:

$$Z = Z_E + Z_R \in \mathbb{R}^{N \times D}$$

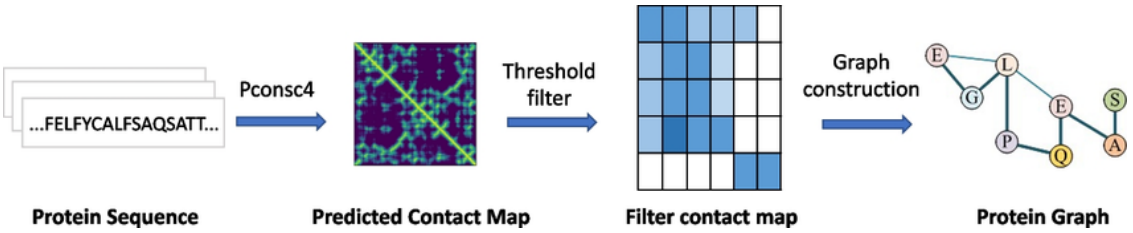

Figure 3.2: Generating a protein graph from sequence information. The amino acid sequence is used to predict a residue-residue contact map (using information from their corresponding PDB entries). This map is then filtered based on a distance threshold (e.g., < 10 Å) to create a binary adjacency matrix, which defines the edges of the final protein graph representation used in downstream analyses.

effectively integrating deep contextual information with explicit amino acid identity features.

3.2.2 Structural Information

To incorporate three-dimensional structural information, we extract atomic coordinates for each residue from the Protein Data Bank (PDB) [65]. These coordinates are used to compute geometric features or define spatial relationships between residues, allowing the model to reason about the protein's physical structure.

3.2.3 Contact Map Construction

Edges in the protein graph are defined using a C_α-based contact map. Specifically, an undirected edge is added between two residues if the Euclidean distance between their C_α atoms is less than 10 Å. The resulting binary contact map serves as the adjacency matrix A for the graph G, capturing local structural neighborhoods critical for protein function.

3.3 ProteinRPN Core Architecture

In this section we discuss the core components responsible for processing the protein graph representation and generating functional predictions.The architecture consists of three primary functional stages: the Region Proposal Network (RPN) module, which identifies and refines candidate functional regions; the Functional Attention Layer, which selectively enhances the features of these identified regions; and the final Prediction Head, which aggregates the

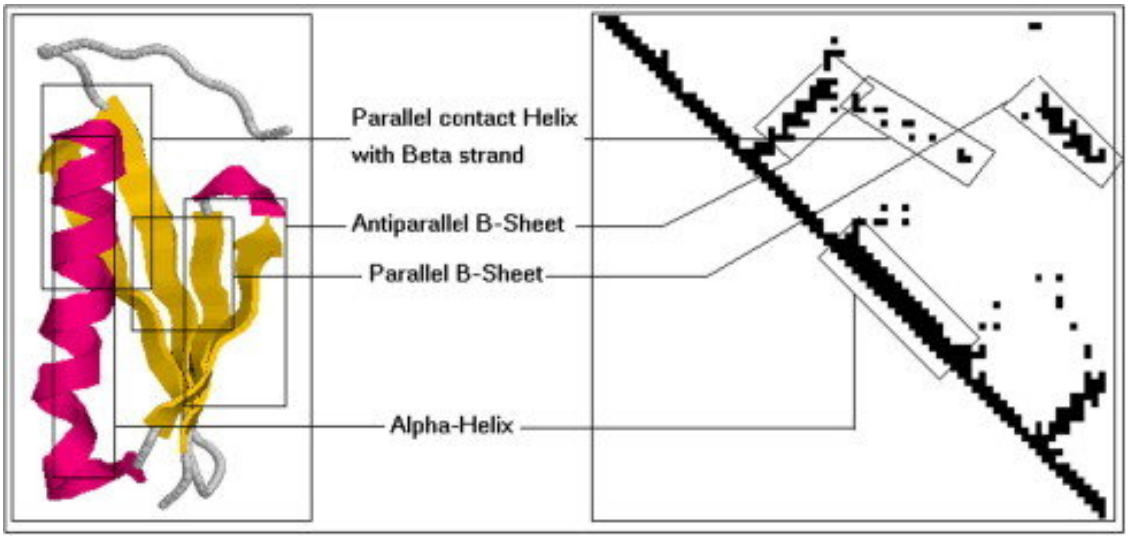

Figure 3.3: Relationship between protein secondary structure elements and contact map patterns. The ribbon diagram (left) shows α-helices and β-sheets, which correspond to characteristic diagonal and off-diagonal patterns in the contact map (right).

refined features and classifies the protein into Gene Ontology (GO) and Enzyme Commission (EC) terms.

3.3.1 Region Proposal Network (RPN) Module

Inspired by object detection methods in computer vision, specifically Region Proposal Networks (RPNs) in Faster R-CNN [14], we introduce a similar idea for protein function prediction on graphs. The core idea is to focus on small regions of the protein graph that are likely to contain functional residues, rather than processing the entire graph uniformly. This is important because, in most proteins, only a small subset of residues are actually involved in carrying out specific functions. To our knowledge, this is the first method to apply graph-based region proposals in the context of protein function prediction.

Graph Convolution Layers

The ProteinRPN model begins its analysis by processing the input protein graph, denoted as $G(Z, A)$, where Z represents the initial node (residue) feature matrix and A is the adjacency matrix capturing residue proximities. To enrich these initial features with local contextual information, the Region Proposal Network (RPN) module employs k layers of

Graph Convolutional Networks (GCNs) [66].

Let $H^{(0)} = Z$ be the initial hidden node embedding matrix. The embeddings are updated iteratively through the GCN layers according to the standard propagation rule:

$$H^{(i+1)} = \text{ReLU}(\tilde{D}^{-0.5}\tilde{A}\tilde{D}^{-0.5}H^{(i)}W^{(i)}) \tag{3.1}$$

where $\tilde{A} = A + I$ is the adjacency matrix with self-loops added, $\tilde{D}$ is the diagonal degree matrix of $\tilde{A}$ used for normalization, $W^{(i)}$ is the trainable weight matrix for layer i, and ReLU is the Rectified Linear Unit activation function.

After k message-passing layers, the resulting node embedding matrix, $Z_1 = H^{(k)} \in \mathbb{R}^{N \times D_1}$ (where N is the number of residues and D_1 is the dimension of the new embeddings), encapsulates information from each node's respective k-hop neighborhood. This effectively extends each node's receptive field to encompass its k-hop subgraph. Consequently, each node v_i in the graph, now represented by its embedding $Z_{1,i}$, can be considered a representative of its corresponding k-hop subgraph, termed an "anchor." This initial processing step transforms the original graph G into a conceptual graph $G'(Z_1, A)$, where each node's features are enriched with information about its local structural environment. Based on empirical results, setting $k = 2$ was found to be sufficient for capturing the local context relevant to identifying functional residues within proteins. These anchor representations Z_1 are then utilized in the subsequent step of the RPN to predict the likelihood of each anchor containing a functionally relevant region.

Anchor Functionality Prediction

The second part of the region proposal module involves predicting whether an anchor region is functionally relevant. We formulate this as a node classification task: for each node v_i in the latent graph $G'(Z_1, A)$, the goal is to predict whether its corresponding k-hop subgraph contains functional residues.

To perform this classification, we apply a Graph Attention Network (GAT) layer [37] on top of the node embeddings Z_1. The GAT layer allows the model to compute attention over neighbors in the latent space, emphasizing structurally or functionally informative regions. The output of this layer provides a functionality score for each anchor, which is used to select a subset of high-confidence regions for further refinement in subsequent steps of the model.

k-Hop Subgraph Extractor

Once the Region Proposal Network generates a set of candidate anchor nodes, each representing a k-hop neighborhood, the next step is to extract these subgraphs in a differentiable and learnable manner. The k-hop Subgraph Extractor module does this by assigning each node a soft importance score, then using these scores to mask the graph and isolate regions most likely to contain functional residues. This module is designed to:

- Enable end-to-end learning by avoiding hard selections.
- Preserve structural context by softly pruning the graph around high-scoring nodes.
- Allow gradient flow for subgraph selection through continuous relaxation.

Initialization The process begins by computing initial importance scores for each node based on its feature representation. This is done using a simple learnable linear layer followed a sigmoid activation:

$$s^{(0)} = \text{sigmoid}(XW + b)$$

Here:

- X is the node feature matrix (from previous modules).
- W is a learnable weight matrix of shape $D \times 1$.
- b is a bias term.

The result is a score vector $s^{(0)} \in [0, 1]^N$, where each element indicates how likely the corresponding node is to be functionally relevant.

k-Hop Diffusion To incorporate information from the local neighborhood of each node, we diffuse the initial scores using a diffusion matrix M:

$$M = \alpha A + (1 - \alpha)I$$

where:

- A is the adjacency matrix of the protein graph.

- I is the identity matrix (adds self-connections).
- α is a diffusion factor controlling how much importance spreads to neighbors.

We then compute the diffused scores $s^{(k)}$ after k hops:

$$s^{(k)} = M^k s^{(0)}.$$

This step ensures that scores are influenced by neighboring residues across k hops, allowing the extractor to highlight localized clusters of potentially functional residues.

Soft Subgraph Extraction With the diffused scores $s^{(k)}$, we construct a soft subgraph by masking both node features and the adjacency matrix: Node features are reweighted:

$$X' = s^{(k)} \odot X$$

where $\odot$ denotes element-wise multiplication, and each node feature vector is scaled by its importance score. Adjacency matrix is softly reconstructed: For edges between different nodes ($i \neq j$):

$$A'_{ij} = s_i^{(k)} \cdot s_j^{(k)} \cdot A_{ij}$$

For self-loops ($i = j$):

$$A'_{ii} = \max((s_i^{(k)})^2 \cdot A_{ii}, A_{ii})$$

This results in a softened view of the graph $G'(X', A')$, where high-scoring nodes and their interactions are preserved while irrelevant regions are downweighted.

Entropy Regularization To encourage the model to make clear decisions about which nodes to keep or discard, we apply an entropy-based regularization loss:

$$\mathcal{L}_{\text{entropy}} = -\frac{1}{N} \sum_{i=1}^{N} [s_i^{(k)} \log(s_i^{(k)}) + (1 - s_i^{(k)}) \log(1 - s_i^{(k)})]$$

This entropy loss penalizes uncertain (mid-range) scores and pushes them toward binary decisions, 0 (irrelevant) or 1 (important). As a result, the model learns to focus on small, coherent clusters of residues most likely to be functionally relevant.

This soft subgraph extraction mechanism ensures that ProteinRPN retains the flexibility of

learning which regions to focus on while still enabling structural and contextual awareness. By isolating likely functional regions and filtering out irrelevant residues, this module ensures that downstream components, like the functional attention and prediction layers, operate on focused and biologically meaningful subgraphs, improving both performance and interpretability.

Node Drop Pooling: Pruning Non-functional Residues

After identifying promising k-hop subgraph anchors around each node, the next challenge is to aggressively prune these regions to isolate only the functionally relevant residues. This is crucial, as most anchor regions still contain many non-functional nodes that can introduce noise and hurt downstream performance. To achieve this, we develop a novel differentiable node drop pooling mechanism. This module learns a soft binary mask over each node, effectively deciding which nodes to retain for further processing and which to discard.

Differentiable Node Drop Pooling This module works by assigning a drop probability to each node in the subgraph, indicating how likely it is to be pruned. It is fully differentiable, allowing gradients to flow through the selection process and enabling end-to-end training.

Node Drop Probability We compute a drop score for each node using a lightweight graph neural network applied to the current graph structure at layer L:

$$D^{(L)} = \text{sigmoid}(\text{GNN}_{\text{drop}}(A^{(L)}, Z^{(L)}))$$

Here:

- $A^{(L)}$ is the current adjacency matrix.
- $Z^{(L)}$ is the current node feature matrix.
- GNN_{drop} is a trainable GNN that outputs a drop score for each node.
- $D^{(L)} \in [0, 1]^N$ contains retention probabilities , a score close to 1 indicates a node should be retained, while scores near 0 indicate it should be dropped.

Domain Knowledge-based Node Scores

Proximity Scores This component is motivated by the observation that functional residues often cluster together in three-dimensional space, even if they are distant in the primary sequence. Instead of merely using a binary contact map, the proximity score P_i for a residue i is calculated by summing the inverse of its distances to all other residues j:

$$P_i = \alpha_{ps} \sum_{j \neq i} \frac{1}{d_{ij}} \tag{3.2}$$

where d_{ij} is the Euclidean distance between the C_α atoms of residues i and j, and α_{ps} is a learnable scaling factor. This formulation prioritizes residues that are tightly clustered with a few other residues over those that might be moderately close to many, aligning with structural characteristics observed in functional sites such as those in the PDBSite dataset [15].

Secondary Structure Scores Experimental evidence indicates that certain types of functional residues, particularly catalytic ones, exhibit a preference for specific secondary structural environments. For instance, catalytic residues are frequently found within α-helices (approximately 39%) and β-sheets (approximately 28%), and are less common in less structured loops or coil regions [67]. To leverage this prior knowledge, ProteinRPN assigns higher predicted scores to residues located within α-helices or β-sheets. The secondary structure assignment for each residue is obtained using DSSP (Dictionary of Secondary Structure in Proteins) [68].

These two domain knowledge-driven scores are combined with scores from GNN_{drop} to produce a final functionality score for each node within the proposed anchors. These scores are then converted into probabilities (e.g., using a sigmoid function), and residues with the highest probabilities are identified as functional, guiding the pruning process within the Node Drop Pooling layer and subsequent feature enrichment by the Functional Attention Layer. This integration of explicit biological priors aims to make the functional region identification more accurate and biologically meaningful.

Feature Update Using the retention probabilities $D^{(L)}$, we update the node features by applying an element-wise multiplication:

$$Z^{(L+1)} = D^{(L)} \odot Z^{(L)}$$

This has the effect of zeroing out the features of nodes with low retention scores, allowing the model to focus only on the most important residues within the subgraph.

Adjacency Matrix Update We similarly update the adjacency matrix using the outer product of the retention scores:

$$A^{(L+1)} = (D^{(L)}(D^{(L)})^T) \odot A^{(L)}$$

This softly removes edges that involve dropped nodes by scaling them down. The resulting matrix $A^{(L+1)}$ is a fractional adjacency matrix where the strength of each edge reflects the retention confidence of the connected nodes.

This node drop pooling layer allows the model to dynamically refine anchor regions, focusing only on clusters of residues with high functional relevance. Unlike hard pruning, the soft probabilistic formulation retains gradient flow, making it learnable and stable during training.

By the end of this module, each subgraph has been reduced to a compact, functionally enriched set of residues, ready for further analysis through surface alignment and functional attention.

3.3.2 Functional Attention Layer

Following the identification of candidate functional residues by the region proposal and refinement stages, their representations are further processed and enriched through a dedicated Functional Attention Layer. The primary purpose of this layer is to selectively amplify the features associated with the predicted functional nodes while retaining contextual information from neighboring non-functional nodes. This is achieved by assigning attention weights to graph edges based on their connectivity to the predicted functional residues, thereby emphasizing relationships deemed critical for protein function. This edge-centric attention mechanism iteratively enhances the functional node representations and helps preserve the structural integrity of the identified functional subgraphs, mitigating the risk of fragmentation

that can occur when selecting individual nodes in isolation.

To implement this, the original residue features $Z \in \mathbb{R}^{N \times D}$ (combining ESM-1b embeddings and amino acid identity embeddings) serve as the input node feature matrix for enrichment. For each edge (i, j) connecting residue i to residue j in the protein graph, an initial attention score e_{ij} is computed. This score is derived from the concatenation of the feature vectors Z_i and Z_j, which is then passed through a linear transformation parameterized by a learnable weight vector $a \in \mathbb{R}^{2D \times 1}$ (where D is the feature dimension), followed by a ReLU activation function to discard negative scores:

$$e_{ij} = \text{ReLU}(a^T[Z_i||Z_j]) \tag{3.3}$$

This initial attention score e_{ij} is then modulated based on whether the target node j was previously identified as functional ($z_j = 1$) or contextual ($z_j = 0$). The adjusted score e'_{ij} is calculated as:

$$e'_{ij} = \alpha_{FA} \cdot e_{ij} \cdot z_j + \beta_{FA} \cdot e_{ij} \cdot (1 - z_j) \tag{3.4}$$

Here, $\alpha_{FA} \geq 1$ and $\beta_{FA} < 1$ are scaling factors. This adjustment mechanism increases the attention paid to edges connected to functional nodes (scaled by α_{FA}) while reducing the attention for edges connected to contextual nodes (scaled by β_{FA}). For the experiments in this work, α_{FA} was set to 1 and β_{FA} was set to 0.5 to explicitly prioritize focus on the identified functional nodes.

The final attention coefficients α_{ij} are obtained by normalizing the adjusted scores e'_{ij} across all neighbors i of the target node j (e.g., using softmax). These coefficients quantify the influence that each neighboring node i exerts on the target node j.

Finally, the updated feature vector Z_{2j} for node j is computed by aggregating the feature vectors Z_i from its neighbors $\mathcal{N}(j)$, weighted by the corresponding attention coefficients α_{ij}, and transformed by a learnable weight matrix $W \in \mathbb{R}^{D_2 \times D}$ (similar to a standard GAT layer):

$$Z_{2j} = \sum_{i \in \mathcal{N}(j)} \alpha_{ij} \cdot W \cdot Z_i \tag{3.5}$$

This aggregation process ensures that subgraphs surrounding the predicted functional residues receive greater attention and contribute more significantly to the final node representations Z_2. This mechanism enhances the model's capacity to capture context-aware interactions crucial

for function, leading to a more comprehensive understanding of the protein's functionally relevant regions.

3.3.3 Prediction Head Module

Graph Multiset Pooling: Aggregating Functional Signals

After functional region proposals have been refined and enriched, the final step of the ProteinRPN pipeline is to aggregate the node-level embeddings into a comprehensive graph-level representation. This representation summarizes the protein as a whole and is used to predict Gene Ontology (GO) and Enzyme Commission (EC) terms.

To achieve this, we employ a Graph Multiset Transformer (GMT) [69], a hierarchical attention-based pooling mechanism that effectively captures both local residue-level interactions and global structural context. Unlike standard global pooling techniques (e.g., mean or max pooling), which treat all nodes equally, GMT leverages learnable super-nodes to selectively summarize the most informative features from the graph.

Learnable Super-Nodes for Global Pooling The GMT module introduces K learnable super-nodes (queries) $Q_{\text{super}} \in \mathbb{R}^{K \times D}$. These super-nodes interact with the graph's node embeddings (output from previous layer, let's call it $H \in \mathbb{R}^{N \times D}$) via multi-head attention to extract meaningful, high-level features from the entire graph. Each super-node is designed to attend to a different aspect of the graph, enabling diverse and informative pooling.

Let H be the input node embeddings and A the graph adjacency matrix. The key and value matrices are computed by applying separate GCN layers (or linear projections):

$$K_{\text{nodes}} = \text{GCN}_1(H, A) \in \mathbb{R}^{N \times D_k}$$
$$V_{\text{nodes}} = \text{GCN}_2(H, A) \in \mathbb{R}^{N \times D_v}$$

The attention scores are computed between the learnable queries Q_{super} and the node key vectors K_{nodes}:

$$C = \text{softmax}\left(\frac{Q_{\text{super}} K_{\text{nodes}}^T}{\sqrt{D_k}}\right) V_{\text{nodes}} \in \mathbb{R}^{K \times D_v}$$

where C represents the updated embeddings for each super-node based on its interactions with the graph nodes.

If using multi-head attention with h heads, where $D = h \times d_{\text{head}}$: Embeddings $C_1, \ldots, C_h$ (each in $\mathbb{R}^{K \times d_{\text{head}}}$) are produced by each head. These are concatenated and passed through a fully connected layer:

$$U = \text{FC}(\text{concat}(C_1, \ldots, C_h)) \in \mathbb{R}^{K \times D}$$

This yields a semantics-aware super-node embedding matrix U, which consolidates the information each head has learned.

Attention-Based Graph Pooling To summarize the super-node embeddings U into a final graph-level representation, we apply a single-head attention pooling step. A global learnable query vector $q_p \in \mathbb{R}^D$ attends over the super-node embeddings U:

$$\begin{aligned} K_p &= UW_K^p \\ V_p &= UW_V^p \\ z &= \text{softmax}\left(\frac{q_p K_p^T}{\sqrt{D}}\right) V_p \in \mathbb{R}^D \end{aligned}$$

where W_K^p, W_V^p are learnable projection matrices, and z is the final pooled graph-level representation.

The GMT layer allows the model to flexibly pool information from only the most relevant residues, as determined by previous modules, while still preserving the broader context of the graph's structure. This ensures that the final representation used for function prediction is both fine-grained and globally informed, a critical requirement for tasks like GO term prediction where local motifs and long-range dependencies both matter.

MLP Prediction Head: GO and EC Term Classification

The final graph-level embedding z, obtained from the Graph Multiset Transformer (GMT), is passed through a multi-layer perceptron (MLP) to produce predictions for Gene Ontology (GO) and Enzyme Commission (EC) terms.

Given the multi-label nature of the task, where each protein may be associated with multiple functional annotations across different subontologies (Biological Process, Molecular Function, Cellular Component), we use sigmoid activation to independently predict the probability of each term.

Formally, the prediction is computed as:

$$\hat{y} = \sigma(\text{MLP}(z))$$

where:

- $z \in \mathbb{R}^D$ is the graph-level representation.
- MLP is a feedforward neural network with one or more hidden layers.
- $\hat{y} \in \mathbb{R}^L$ is the predicted vector of probabilities for L possible functional terms (GO or EC).
- σ denotes the element-wise sigmoid function.

Each value $\hat{y}_i \in [0, 1]$ represents the likelihood that the protein is associated with the i^{th} term.

3.4 Optimization Framework

To enable accurate functional region identification and robust protein function prediction, ProteinRPN is trained using a two-phase loss framework. During pretraining, the model focuses on learning to identify functional residues at the node-level. In the downstream phase, it predicts GO and EC terms at the graph-level while enforcing structural coherence in the predicted functional subgraphs.

3.4.1 Pretraining Losses: Functional Residue Identification

The pretraining phase supervises the identification of functional residues using residue-level labels derived from structural annotations.

To promote smooth predictions across connected nodes, we use a cluster-wise agreement loss, defined over each proposed functional subgraph $G_{\text{sub}} = (V_{\text{sub}}, E_{\text{sub}})$:

$$\mathcal{L}_{\text{cluster}} = \sum_{(i,j) \in E_{\text{sub}}} A_{ij}(D_i - D_j)^2$$

Here D_i represents the predicted importance score for node i. This term penalizes large differences in predicted importance scores between adjacent nodes within the proposed subgraph, encouraging structural consistency and smoothness.

To evaluate the quality of predicted subgraphs, we introduce a subgraph coverage loss based on Intersection over Union (IoU). This loss measures the overlap between the predicted residue importance scores $D \in [0,1]^N$ and binary ground truth labels $Y \in \{0,1\}^N$. The intersection and union are computed as:

$$\text{Intersection} = \sum_{i=1}^{N} D_i Y_i$$

$$\text{Union} = \sum_{i=1}^{N} D_i + \sum_{i=1}^{N} Y_i - \text{Intersection}$$

The IoU loss is then formulated as:

$$\mathcal{L}_{\text{IoU}} = 1 - \frac{\text{Intersection}}{\text{Union} + \varepsilon}$$

where ε is a small constant for numerical stability. This term penalizes both false positives and false negatives, encouraging the model to produce subgraphs that closely match the ground truth functional annotations.

Finally, to promote sharp, confident predictions at the node-level, we apply an entropy regularization loss on the predicted scores D_i:

$$\mathcal{L}_{\text{entropy}} = -\frac{1}{N} \sum_{i=1}^{N} [D_i \log D_i + (1 - D_i) \log(1 - D_i)]$$

This regularizer pushes the predicted values D_i closer to binary outcomes (0 or 1), discouraging indecisive or overly smooth score distributions.

The total pretraining loss combines these terms: $\mathcal{L}_{\text{pretrain}} = \mathcal{L}_{\text{IoU}} + \lambda_1 \mathcal{L}_{\text{cluster}} + \lambda_2 \mathcal{L}_{\text{entropy}}$, where λ_i are weighting hyperparameters.

3.4.2 Downstream Losses: GO and EC Term Prediction

In the downstream phase, the model is optimized to predict protein function using a graph-level embedding z derived from the Graph Multiset Transformer (GMT). The main objective

is a multi-label binary cross-entropy (BCE) loss over T terms for a batch of B proteins:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{B \cdot T} \sum_{i=1}^{B} \sum_{j=1}^{T} [y_{ij} \log(\hat{y}_{ij}) + (1 - y_{ij}) \log(1 - \hat{y}_{ij})]$$

Here, y_{ij} and $\hat{y}_{ij}$ denote the ground truth and predicted presence of term j for protein i.

To improve representation learning, we incorporate a contrastive loss $\mathcal{L}_{\text{con}}$ that combines supervised contrastive (SupCon) [42] and self-supervised InfoNCE [43] objectives. The SupCon term brings together representations z_i, z_j of proteins that share at least one GO term (indicated by $y_i \cap y_j \neq \emptyset$):

$$\mathcal{L}_{\text{SupCon}} = -\frac{1}{B} \sum_{i=1}^{B} \sum_{j \neq i} \mathbf{1}\{y_i \cap y_j \neq \emptyset\} \cdot \log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k \neq i} \exp(\text{sim}(z_i, z_k)/\tau)}$$

The InfoNCE term encourages consistency between original (z_i) and perturbed embeddings (z_i') of the same protein:

$$\mathcal{L}_{\text{InfoNCE}} = -\frac{1}{B} \sum_{i=1}^{B} \log \frac{\exp(\text{sim}(z_i, z_i')/\tau)}{\sum_{j=1}^{B} \exp(\text{sim}(z_i, z_j')/\tau) + \sum_{j \neq i} \exp(\text{sim}(z_i, z_j)/\tau)}$$

Here, $\text{sim}(\cdot, \cdot)$ is cosine similarity, and τ is a temperature parameter. These losses are combined: $\mathcal{L}_{\text{con}} = \alpha_{\text{SupCon}} \mathcal{L}_{\text{SupCon}} + \alpha_{\text{InfoNCE}} \mathcal{L}_{\text{InfoNCE}}$.

To ensure that the predicted functional regions form spatially coherent clusters, we introduce a connected components penalty. This loss penalizes predicted subgraphs $G_i'' = (V_f, E_f)$ induced by the predicted functional residues in protein i that contain multiple disconnected components $\kappa(G_i'')$:

$$\mathcal{L}_{\text{penalty}} = \alpha_{\text{cc}} \cdot \frac{1}{B} \sum_{i=1}^{B} \frac{\kappa(G_i'')}{\max(1, \sum_{j=1}^{T} y_{ij})}$$

where α_{cc} is a weighting term. More details on implementation of the loss term can be found in the Appendix.

The final downstream loss function is a weighted combination of all components:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{con}} + \mathcal{L}_{\text{penalty}}$$

This formulation ensures that the model not only makes accurate functional predictions but also learns compact, biologically meaningful subgraphs aligned with the structural and

functional organization of proteins.

Chapter 4

Experimental Setup

In this chapter, we dive deep into the experimental setup followed to demonstrate the effectiveness of our framework ProteinRPN, detailing the experimental design, the datasets used, the training procedure and the comparative analysis performed to validate the model against the state-of-the-art protein function prediction benchmarks. We start by covering the pretraining and downstream task datasets, followed by model training strategy, evaluation metrics used, baseline models we compare against, and finally, the ablation experiments.

4.1 Datasets

In order to pretrain the region proposal module in our framework, which detects and annotates functional residue clusters in the protein, we pretrain it on a combination of protein and enzyme active site annotated datasets such as PDBSite, BioLiP, ScanNet and DeepProSite. This is followed by downstream training on the CAFA datasets prepared by Gu et al. [34] for GO term prediction and the standard EC benchmark from GearNet [49] for EC term prediction.

4.1.1 Regional Proposal Module Pretraining Dataset

The region proposal component was pretrained using functional site annotations to learn to identify relevant residue clusters. For this, we use the PDBSite dataset [15]. The dataset is derived from the Protein Data Bank (PDB) [65] and contains comprehensive annotations

of biologically active sites. The version used contains 4723 active sites associated with 197 different functions across 603 proteins. PDBSite was crucial for pretraining due to its diversity in functional categories and explicit residue-level annotation of functional sites, making it suitable for training the region proposal module.

4.1.2 Downstream Task Datasets

GO Prediction Benchmark (HEAL Dataset)

Our primary dataset for Gene Ontology (GO) term prediction is based on the benchmark curated by Gu et al. [34] for evaluating the HEAL model. This dataset is derived from the widely used DeepFRI [12] dataset, ensuring a standardized basis for fair comparison.

The dataset includes protein sequences from two sources: 36 629 sequences from the PDB and 42 994 sequences from the AlphaFold Database (AFDB). Representative PDB chains with existing functional annotations and high-resolution structures were selected. Redundancy was managed by clustering sequences at 95% sequence identity.

Experimentally solved 3D structures from the PDB were used to construct protein residue graphs. GO term annotations were obtained from the SIFTS resource [70], applying strict criteria for annotation transfer (e.g., $\geq$90% sequence identity and $\geq$70% coverage between PDB models and UniProtKB sequences) to ensure reliability. The final label space covered 489 Molecular Function (MF) terms, 1943 Biological Process (BP) terms, and 320 Cellular Component (CC) terms. A subset of 44 137 proteins from the AFDB associated with low-frequency GO terms (Information Content, IC $>$ 10 in the training set) was used for specific analyses, with structures predicted by AlphaFold2 [29]. The Information Content $IC(f)$ of a term f is often defined based on its frequency in the annotation database:

$$IC(f) = -\log_2 P(f)$$

where $P(f)$ is the probability (frequency) of term f appearing in the annotations.

The complete non-redundant dataset was partitioned into training, validation, and test sets (approx. 80%/10%/10%). The test set was fixed across evaluations and comprised only proteins with experimental PDB structures and experimentally verified GO annotations (evidence codes EXP, IDA, etc.). This standardized test set includes proteins with varying sequence identity (30% to 95%) relative to the training set. Each test set protein has at least

one experimentally confirmed GO term in each ontology (MF, BP, CC).

EC Prediction Benchmark

For Enzyme Commission (EC) number prediction, we used the dataset established by Gligorijevic et al. [12], which employs multi-cutoff sequence identity splits (e.g., test set sequences having $\leq 95\%$ identity to training set sequences). Enzyme classes were retrieved from levels 3 and 4 of the EC hierarchy, resulting in 538 distinct EC classes used as labels.

4.2 Evaluation Metrics

To rigorously assess the performance of ProteinRPN and compare it with baseline methods, we use standard evaluation metrics widely employed in the CAFA community for protein function prediction tasks [6].

As noted in Chapter 2, raw model predictions (probability vectors) must be post-processed to respect the hierarchical structure of GO and EC ontologies according to the True Path Rule. If a specific term (child) is predicted positive (above a threshold τ), the positive prediction is propagated up to all its ancestor terms. Conversely, if a general term (parent) is predicted negative, the negative prediction is propagated down to all its descendant terms. This ensures hierarchical consistency.

4.2.1 Protein-centric $F_{\max}$

The primary metric is protein-centric maximum F1-score ($F_{\max}$). For each protein i, the F1 score is calculated at various prediction thresholds $\tau \in [0, 1]$. The $F_{\max}$ is the maximum F1 score achieved across all thresholds, averaged over all proteins in the evaluation set. Let $P_i(\tau)$ be the set of terms predicted for protein i at threshold τ, and T_i be the set of true terms for protein i. The precision $pr_i(\tau)$ and recall $rc_i(\tau)$ for protein i are:

$$pr_i(\tau) = \frac{|P_i(\tau) \cap T_i|}{|P_i(\tau)|}$$
$$rc_i(\tau) = \frac{|P_i(\tau) \cap T_i|}{|T_i|}$$

The F1 score for protein i at threshold τ is:

$$F_{1,i}(\tau) = \frac{2 \cdot pr_i(\tau) \cdot rc_i(\tau)}{pr_i(\tau) + rc_i(\tau)} \quad (\text{if } pr_i(\tau) + rc_i(\tau) > 0, \text{ else } 0)$$

The $F_{\max}$ for protein i is the maximum F1 score across all thresholds:

$$F_{\max,i} = \max_{\tau \in [0,1]} \{F_{1,i}(\tau)\}$$

The final reported $F_{\max}$ is the average over all N_{proteins} in the test set:

$$F_{\max} = \frac{1}{N_{\text{proteins}}} \sum_{i=1}^{N_{\text{proteins}}} F_{\max,i}$$

A higher $F_{\max}$ indicates better overall performance.

4.2.2 Semantic Distance ($S_{\min}$)

The Semantic Distance ($S_{\min}$) metric evaluates predictions based on their position within the GO hierarchy, using the Information Content (IC) of terms. Lower $S_{\min}$ values indicate better performance (predictions are closer to the true terms in the ontology). It is calculated as the minimum Euclidean distance in the (misinformation, remaining uncertainty) plane across thresholds τ:

$$S_{\min} = \min_{\tau \in [0,1]} \sqrt{ru(\tau)^2 + mi(\tau)^2}$$

where the average remaining uncertainty $ru(\tau)$ and misinformation $mi(\tau)$ over N_{proteins} are:

$$ru(\tau) = \frac{1}{N_{\text{proteins}}} \sum_{i=1}^{N_{\text{proteins}}} \sum_{f \in T_i \setminus P_i(\tau)} IC(f)$$

$$mi(\tau) = \frac{1}{N_{\text{proteins}}} \sum_{i=1}^{N_{\text{proteins}}} \sum_{f \in P_i(\tau) \setminus T_i} IC(f)$$

4.2.3 Area Under the Precision-Recall Curve (AUPR)

The Area Under the Precision-Recall Curve (AUPR) is calculated by integrating precision over recall across all possible thresholds. AUPR is particularly useful for imbalanced datasets,

as is common with GO/EC terms where many terms annotate few proteins. It focuses on the model's ability to correctly identify positive instances. We report macro-AUPR (average AUPR across all terms, treating each term equally) and micro-AUPR (AUPR calculated from globally summed true positives, false positives, etc., weighting by term frequency). Higher AUPR indicates better performance, especially for rare terms.

4.3 Baseline Methods

To demonstrate the performance of ProteinRPN, we compare it against state-of-the-art protein function prediction models spanning different categories:

- **Sequence-based:** BLAST [71], DeepGO [55].
- **Protein Language Model (PLM) based:** ProtT5 [40], ESM-2 [39], SaProt [41].
- **Sequence and Structure-based:** FunFams [72], DeepFRI [12], HEAL [34], TAWFN [73], DeepGO-SE [74].

Results for top-performing baselines were reproduced using their published code and recommended settings on our specific dataset splits where possible. For others, results were taken from prior publications (e.g., [34]) that used the same benchmark datasets and evaluation protocols.

Sequence-Based Methods

- **BLAST [71]:** A foundational sequence alignment tool. Function is transferred from the best hit in the training set to a test sequence, typically using sequence identity or E-value (e.g., threshold of 10^{-3}) to weight or select annotations.
- **DeepGOPlus [55, 56]:** Combines sequence homology scores from DiamondBLAST with predictions from a 1D CNN, using a tuned weight based on validation performance.
- **DeepGO-SE [74]**: Predicts Gene Ontology (GO) functions from protein sequences by integrating a pretrained large language model with a neuro-symbolic framework that leverages GO axioms, framing protein function prediction as a task of approximate semantic entailment.

PLM-Based Baselines

These methods use large pre-trained Protein Language Models (PLMs) to generate embeddings, which are then fed into a simple prediction head (e.g., MLP) fine-tuned for the task.

- **ProtT5 [40]:** An encoder-decoder transformer PLM (3B parameters). Weights frozen, MLP head trained.
- **ESM-2 [39]:** An encoder-only transformer PLM (650M parameter version used). Weights frozen, MLP head trained.
- **SaProt [41]:** A structure-aware PLM integrating sequence and predicted structure (650M parameter version used). Weights frozen, MLP head trained.

Sequence and Structure-Based Methods

- **FunFams [72]:** A domain-based method using CATH functional families. Annotations are transferred based on the best-matching domain found via sequence/structure search.
- **DeepFRI [12]:** One of the first GCN-based models combining sequence embeddings (from a PLM) with structural information (from predicted contact maps).
- **HEAL [34]:** The main baseline, using a hierarchical graph transformer with contrastive learning to capture global and local structural features.
- **TAWFN [73]:** A Two-model Adaptive Weight Fusion Network combining CNNs (sequence) and GCNs (structure) with adaptive feature fusion. Retrained on our datasets.

4.4 Experimental Design and Ablation Studies

We systematically evaluate the contribution of key components of our ProteinRPN framework and training strategy to overall performance through ablation studies. We removed or modified specific components, retrained the model, and compared performance against the full model using the standard metrics ($F_{\max}$, $S_{\min}$, AUPR) on the test set. Specifically, we test the impact of:

- Removing the region proposal module entirely.
- Domain Knowledge ablation (e.g., removing surface alignment or specific pretraining losses).
- Removing Contrastive Learning (ProteinRPN vs. ProteinRPN w/o CL).
- Combined ablation (ProteinRPN w/o Domain Knowledge and w/o CL).
- Replacing the Graph Multiset Transformer pooling with a more advanced alternative like Polynormer [75].
- Replacing the combined InfoNCE + SupCon loss with SINCERE [76].

The results of these ablation experiments are reported and discussed in Chapter 5.

4.4.1 Functional Residue Visualization

To qualitatively assess the performance of the region proposal module, we visualize the predicted functional residues on selected proteins from the PDBSite test set (e.g., PDB IDs 2BCC chain B, 2CHG chain A). We analyze the number of predicted functional residues, the number correctly identified compared to ground truth, and the spatial clustering of the predictions. This provides insight into the module's ability to identify functionally relevant residues, validating the core hypothesis behind ProteinRPN.

4.5 Implementation and Training Details

This section provides details necessary for reproducing the training process.

- **Hardware:** Training was performed on a single NVIDIA A100 GPU with 40 GB of memory, alongside a system with 256 GB RAM.
- **Software:** Key libraries included PyTorch and PyTorch Geometric.
- **Training Parameters:** The Adam optimizer was used with a learning rate of 1×10^{-4} and a batch size of 48. Models were trained for up to 100 epochs.

- **Early Stopping:** To prevent overfitting, an early stopping criterion was employed with a patience of five epochs, monitoring performance on the validation set (e.g., maximizing validation $F_{\max}$ or minimizing validation loss).
- **Loss Function Weights:** For the combined contrastive loss $\mathcal{L}_{\text{con}} = \alpha_{\text{SupCon}}\mathcal{L}_{\text{SupCon}} + \alpha_{\text{InfoNCE}}\mathcal{L}_{\text{InfoNCE}}$, weights were set to $\alpha_{\text{SupCon}} = 0.1$ and $\alpha_{\text{NCE}} = 0.1$.
- **Key Hyperparameters:** Contrastive loss temperature $\tau = 0.7$; embedding dimensions ($D = 512$); number of GCN layers in RPN ($k = 2$).

Typical training times were approximately 12 hours per model using the specified hardware and batch size.

In this chapter, we described the experimental setup, highlighting the datasets, the multi-stage training approach, the comprehensive evaluation metrics, and the comparative analyses performed. In the next chapter, we discuss the results obtained from implementing these procedures and analyze their implications for the advancement of protein function prediction.

Chapter 5

Results and Discussion

In this chapter we discuss the evaluation of the Protein Region Proposal Network (ProteinRPN) model developed in this thesis. We assess its effectiveness in predicting protein function, specifically focusing on Gene Ontology (GO) term prediction and preliminary results for Enzyme Commission (EC) number prediction, benchmarking its performance against current state-of-the-art methods. The evaluation consists of quantitative analyses using standard CAFA metrics, ablation studies to determine the contribution of key architectural components, and qualitative assessments of the model's ability to identify functionally relevant structural regions via its novel region proposal mechanism. The results are discussed in the context of existing challenges and related work, highlighting the contributions and potential limitations of the ProteinRPN framework.

5.1 Protein Function Prediction Performance (GO Terms)

The primary evaluation of ProteinRPN focused on predicting GO terms using the established HEAL benchmark dataset, ensuring comparability with leading methods. The performance results, summarized in Table 5.1, clearly show that ProteinRPN consistently outperforms the selected baseline methods across all three GO domains (Biological Process - BP, Cellular Component - CC, Molecular Function - MF). This includes outperforming sequence-based methods (BLAST [71], FunFams [72], DeepGOPlus [56], DeepGO-SE [74]), recent sequence/structure-based models (DeepFRI [12], PFresGO [77], HEAL [34], TAWFN [73]), and Protein Language Model-based (PLM) baselines where an MLP head was attached to the PLM embeddings. ProteinRPN achieves the highest scores on the primary $F_{\max}$ metric, as well as the secondary

AUPR and $S_{\min}$ metrics for all domains.

Quantifying these improvements reveals significant gains, particularly over the strong HEAL baseline. ProteinRPN achieves $F_{\max}$ increases of 6.4% for BP, 2.7% for CC, and 6.5% for MF compared to HEAL. Its advantage is further highlighted by improvements in AUPR (15.4% for BP, 8.5% for CC, 1.3% for MF vs. HEAL) and $S_{\min}$ (reductions of 1.8% for BP, 0.9% for CC, 9.2% for MF vs. HEAL) as we can see in Table 5.1.

Rigorous statistical analysis using paired t-tests across three independent runs confirmed that ProteinRPN's performance improvements over HEAL and TAWFN (two advanced baselines) are statistically significant ($p < 0.05$) for the majority of metrics, particularly for $F_{\max}$ and $S_{\min}$ of BP.

The enhanced performance of ProteinRPN is primarily attributed to its unique architecture that proactively identifies and focuses on functionally relevant structural regions. The integrated Region Proposal Network (RPN) module distinguishes ProteinRPN from methods that lack explicit functional site identification mechanisms or rely on post-hoc interpretability techniques like Grad-CAM. Such post-hoc methods often highlight numerous scattered residues, lacking the specificity and direct biological insight provided by ProteinRPN's functional region-centric approach. By refining proposed regions using domain knowledge (secondary structure, proximity) and hierarchy-aware attention, ProteinRPN likely achieves a more accurate mapping between functionally important local structures and overall protein function.

While PLMs like ESM-2, ProtT5, and SaProt provide strong sequence-based representations, supplementary experiments feeding their frozen embeddings into simple MLP classifiers yielded performance considerably lower than ProteinRPN (see Table 5.1). This suggests that ProteinRPN's structure-aware region proposal approach offers complementary benefits, particularly for BP and MF predictions where local structural context is often crucial, effectively leveraging structural information beyond what is captured by sequence embeddings alone. With regards to Enzyme Commision (EC) number prediction, preliminary results acheived an F_{max} of 0.895 for top-level EC class prediction and 0.61 for hierarchical classification.

Table 5.1: Performance comparison of ProteinRPN against baseline methods on the HEAL benchmark dataset for GO term prediction. Methods are grouped by primary input type. Best results in each column are highlighted in bold. Higher F_{max}/AUPR is better; lower S_{min} is better.

Type	Method	F_{max} (↑)			AUPR (↑)			S_{min} (↓)		
		BP	CC	MF	BP	CC	MF	BP	CC	MF
Sequence (Homology)	Blast	0.336	0.448	0.328	0.067	0.097	0.136	0.651	0.628	0.632
Sequence (Domain/Homology)	FunFams	0.500	0.572	0.627	0.260	0.288	0.367	0.579	0.503	0.531
Sequence + Structure	DeepFRI	0.540	0.625	0.613	0.261	0.274	0.495	0.543	0.527	0.437
	HEAL	0.581	0.673	0.708	0.298	0.415	0.630	0.504	0.462	0.369
	DeepGO	0.493	0.577	0.594	0.182	0.263	0.391	0.577	0.550	0.472
	DeepGO-SE	0.566	0.636	0.654	0.233	0.423	0.495	0.530	0.481	0.435
Hybrid (Seq+Ontology)	PFresGO	0.568	0.674	0.692	0.293	0.361	0.602	0.535	0.498	0.417
Hybrid (Seq+Attention Fusion)	TAWFN	0.548	0.609	0.711	0.279	0.346	0.674	0.561	0.539	0.393
Sequence (PLM + MLP)	ProtT5 + MLP	0.327	0.623	0.511	0.056	0.284	0.294	0.637	0.526	0.552
	ESM-2 + MLP	0.351	0.633	0.531	0.062	0.282	0.298	0.622	0.507	0.563
Seq+Struct Embed (PLM + MLP)	SaProt + MLP	0.374	0.506	0.287	0.034	0.037	0.018	0.643	0.612	0.662
Sequence + Structure	**ProteinRPN**	**0.618**	**0.691**	**0.754**	**0.344**	**0.459**	**0.683**	**0.495**	**0.458**	**0.335**

5.2 Module Effectiveness (Ablation Study Results)

To analyze the contributions of ProteinRPN's key architectural components, comprehensive ablation studies were performed. These studies systematically removed specific modules and evaluated the impact on overall function prediction performance, demonstrating the necessity of each component for optimal results.

We start by removing the region proposal mechanism to assess its benefit. This was followed by looking into the effectiveness of specific and precise region proposals via hyperparameter tuning. We follow this up by testing the efficacy of incorporating domain knowledge scoring in the node refinement process and the importance of the contrastive learning paradigm.

Removing the entire Region Proposal mechanism ("w/o Region Proposals") resulted in the most substantial performance degradation across all metrics and GO domains, as shown in Table 5.2. For instance, $F_{\max}$ dropped from 0.618 to 0.581 for BP, 0.691 to 0.673 for CC, and 0.754 to 0.708 for MF. This significant reduction highlights that the core strategy of proactively identifying functional regions is fundamental to the model's success.

Similarly, removing the domain knowledge scoring (secondary structure preference and spatial proximity heuristics) from the node drop pooling layer ("w/o domain knowledge scoring") also led to a decrease in performance (Table 5.2), albeit smaller than removing the entire region proposal system. This result validates the utility of incorporating these biologically-informed heuristics to refine the selection of functionally relevant residues within the proposed regions. The ablation study also highlighted the integral nature of the functional attention layer; removing it would break the information flow, preventing evaluation, indicating its essential role in processing the features identified by the preceding modules. These results collectively confirm that each component contributes meaningfully to the overall performance, with the proactive region proposal strategy being particularly impactful.

In addition to removing entire modules, further experiments tested the sensitivity of ProteinRPN's performance to the parameters governing the region proposal process itself. Specifically, the threshold used to select the final functional residues from the high-scoring anchors was varied. The original model implicitly used a threshold selecting roughly the top 25% of residues based on retention probability within high-confidence anchors. When this threshold was relaxed to select the top 60% of residues, a significant performance drop was observed across all metrics, as detailed in Table 5.3. For example, $F_{\max}$ decreased to 0.595 (BP), 0.672 (CC), and 0.691 (MF). This demonstrates that the performance is sensitive to this

Table 5.2: Ablation study results comparing the original ProteinRPN with versions lacking the Region Proposal mechanism or domain knowledge scoring. Best performance highlighted in bold. Higher F_{max}/AUPR is better; lower S_{min} is better.

Method	F_{max} (↑)			AUPR (↑)			S_{min} (↓)		
	BP	CC	MF	BP	CC	MF	BP	CC	MF
Original ProteinRPN	**0.618**	**0.691**	**0.754**	**0.344**	**0.459**	**0.683**	**0.495**	**0.458**	**0.335**
w/o Region Proposals	0.581	0.673	0.708	0.298	0.415	0.630	0.504	0.462	0.369
w/o domain knowledge scoring	0.611	0.685	0.749	0.342	0.459	0.677	0.498	0.457	0.342

threshold and that simply proposing regions is insufficient; the selectivity in identifying the most probable functional residues within those regions is crucial for achieving high accuracy.

To evaluate the contribution of key methodological components within ProteinRPN, particularly the representation learning strategy and the use of domain knowledge in region refinement, further ablation studies were performed. We also tested for synergistic effects between the presence of contrastive learning and the domain knowledge scoring. These experiments systematically removed the contrastive learning (CL) framework (comprising SupCon and InfoNCE losses) and the domain knowledge scoring based on secondary structure and proximity (DK). The results of these ablations are presented in Table 5.4. Removing the contrastive learning component ("ProteinRPN w/o CL") while keeping the domain knowledge scoring resulted in a moderate decrease in performance across all GO domains and metrics compared to the full ProteinRPN model. For instance, $F_{\max}$ dropped slightly, while AUPR showed a more noticeable decrease, particularly for BP (0.344 to 0.322) and MF (0.683 to 0.648). When domain knowledge scoring (DK) was removed but contrastive learning was kept ("ProteinRPN w CL w/o DK"), performance also decreased compared to the full model, although the impact was generally smaller than removing CL alone, especially for $F_{\max}$ scores. The most significant performance reduction occurred when both contrastive learning and domain knowledge scoring were removed ("ProteinRPN w/o DK w/o CL").

These ablation results confirm the positive contribution of both the contrastive learning framework and the incorporation of domain knowledge to ProteinRPN's overall performance. The performance decrease upon removing contrastive learning underscores its importance in learning robust and discriminative graph representations that likely generalize better for the function prediction task. Similarly, the benefit derived from the domain knowledge scoring highlights the value of guiding the model's attention and refinement process using biologically relevant heuristics. The synergistic effect observed when both components are included justifies their integration into the final ProteinRPN architecture.

5.3 Functional Region Identification and Visualization

Beyond quantitative GO term prediction, we also tested ProteinRPN's ability to identify functionally relevant residues in test proteins from PDBSite. During pretraining on the combined pretraining dataset, the region proposal module demonstrated strong predictive capability, achieving an ROC score of 0.95 for predicting whether an anchor region contained

Table 5.3: Hyperparameter sensitivity analysis for the region proposal selection threshold. Compares original model threshold (25%) with a relaxed threshold (60%). Best performance highlighted in bold. Higher F_{max}/AUPR is better; lower S_{min} is better.

Method (Threshold)	F_{max} (↑)			AUPR (↑)			S_{min} (↓)		
	BP	CC	MF	BP	CC	MF	BP	CC	MF
Original (25%)	**0.618**	**0.691**	**0.754**	**0.344**	**0.459**	**0.683**	**0.495**	**0.458**	**0.335**
Experiment (60%)	0.595	0.672	0.691	0.314	0.425	0.636	0.518	0.472	0.386

Table 5.4: Ablation study results assessing the impact of Contrastive Learning (CL) and Domain Knowledge/Secondary Structure scoring (DK). Best performance highlighted in bold. Higher F_{max}/AUPR is better; lower S_{min} is better.

Model Configuration	F_{max} (↑)			AUPR (↑)			S_{min} (↓)		
	BP	CC	MF	BP	CC	MF	BP	CC	MF
ProteinRPN (Original w CL w DK)	**0.618**	**0.691**	**0.754**	**0.344**	**0.459**	**0.683**	**0.495**	**0.458**	**0.335**
ProteinRPN w DK w/o CL	0.601	0.688	0.741	0.322	0.417	0.648	0.506	0.459	0.356
ProteinRPN w CL w/o DK	0.611	0.689	0.750	0.343	0.459	0.678	0.498	0.458	0.342
ProteinRPN w/o CL w/o DK	0.597	0.680	0.736	0.316	0.424	0.645	0.509	0.467	0.355

functional residues and 0.85 for the subsequent pruning task (node drop pooling). Furthermore, evaluating the final model on a held-out test set of 120 proteins from PDBSite showed high accuracy (97%) in detecting specific functional residues, comparing favorably to other methods, along with a strong AUCROC of 0.8821 for functionality probability predictions across all residues.

Qualitative visualizations provide further insight into the functional residue detection capability of the model. As shown in Figure 5.1 for example proteins 2BCC-B and 2CHG-A, ProteinRPN successfully identifies the majority of experimentally known functional residues (highlighted in green). Importantly, the correctly predicted functional residues tend to form spatially coherent clusters within the proposed regions, aligning with biological understanding of active sites and binding interfaces. While some known functional residues distant from the main cluster were occasionally missed (highlighted in red), the overall results demonstrate the model's strong ability to pinpoint functionally important constellations of residues directly from the structure. This proactive localization offers enhanced interpretability compared to post-hoc methods and directly supports the hypothesis that focusing on these specific regions is key to the model's improved function prediction accuracy.

Table 5.5: Quantitative performance of functional residue prediction on PDBSite test set.

Metric	Value
Anchor Prediction ROC	0.95
Pruning Prediction ROC	0.85
Residue Detection Accuracy	97%
Residue Prediction AUCROC	0.8821

To further investigate the specific choices made within ProteinRPN's architecture, key components were substituted with alternative methods proposed in recent literature: the SINCERE contrastive loss objective [76] (an integrated combination of InfoNCE and Supervised Contrastive loss) and the Polynormer graph transformer [75]. The results of these substitutions are shown in Table 5.6.

Replacing the original combined SupCon and InfoNCE contrastive loss with the unified SINCERE loss [76] resulted in a moderate decrease in performance across $F_{\max}$, AUPR, and $S_{\min}$ metrics (Table 5.6). This suggests that the explicit, potentially more tunable combination of SupCon and InfoNCE used in ProteinRPN provided a better framework for learning robust representations for this specific task compared to the SINCERE formulation.

Substituting the Graph Multiset Transformer (GMT) pooling layer with the Polynormer

Table 5.6: Performance comparison of ProteinRPN variants substituting contrastive loss (SINCERE) or graph transformer (Polynormer). Best results in each column are highlighted in bold. Higher F_{max}/AUPR is better; lower S_{min} is better.

Model Variant	F_{max} (↑)			AUPR (↑)			S_{min} (↓)		
	BP	CC	MF	BP	CC	MF	BP	CC	MF
ProteinRPN CL (Original)	**0.6175**	**0.6906**	**0.7542**	**0.3438**	**0.4527**	**0.6833**	**0.4948**	**0.4576**	**0.3350**
ProteinRPN w SINCERE	0.5823	0.6676	0.7513	0.3128	0.3990	0.6833	0.5180	0.4779	0.3461
ProteinRPN w Polynormer	0.5102	0.6269	0.5877	0.2012	0.3076	0.4030	0.5629	0.5257	0.4908

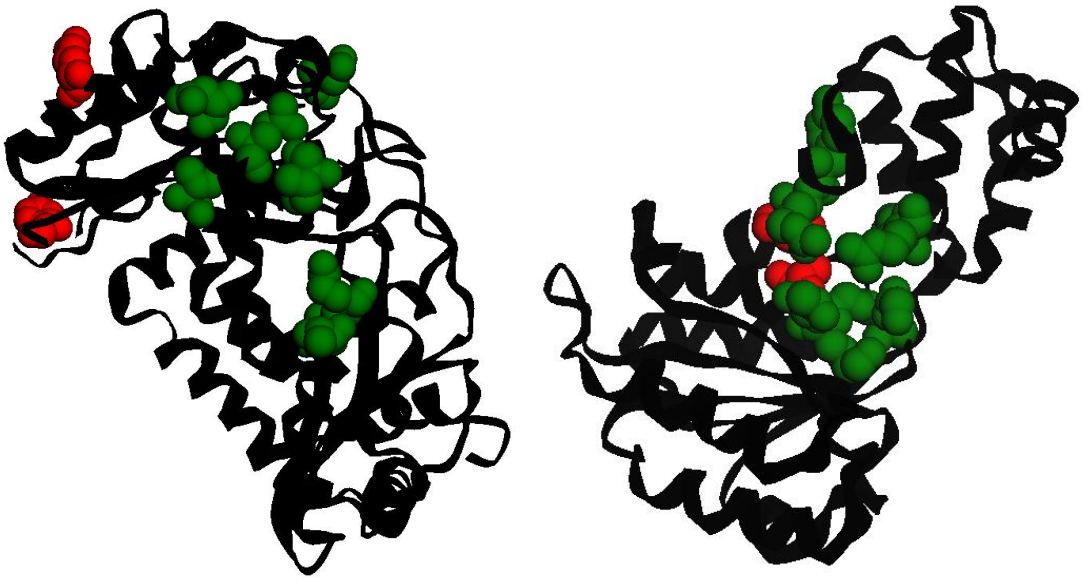

Figure 5.1: Visualization of predicted functional residues by ProteinRPN on example proteins (e.g., *2BCC-B*, *2CHG-A*). Known functional residues correctly identified are shown in green, missed residues in red, predicted residues highlighted.

graph transformer [75] led to a more significant drop in performance across all metrics (Table 5.6). While Polynormer offers high expressivity with linear complexity, its standard pooling mechanism appears less effective for this task than GMT's approach. The use of learnable super-nodes in GMT seems particularly adept at capturing functionally relevant substructures within the protein graph, which is critical for accurate function prediction and justifies its use in the ProteinRPN architecture.

5.4 Summary of Discussion

The results presented in this chapter validate the effectiveness of ProteinRPN as a novel approach to protein function prediction. The model consistently achieves state-of-the-art accuracy on standard benchmarks, significantly outperforming existing methods across various metrics and GO domains. Ablation studies and module comparisons confirm that this strong performance is driven by the synergistic integration of its key components: the proactive graph-based region proposal network, the incorporation of domain knowledge and hierarchy-aware attention for refinement, and the use of contrastive learning for robust representation learning. Furthermore, the model demonstrates a promising ability to accurately localize functionally important residues within the protein structure, offering enhanced biological

interpretability.

In the context of related work, ProteinRPN addresses several identified gaps. By explicitly modeling and identifying functional regions prior to final prediction, it addresses the limitations of methods relying solely on global sequence/structure features or post-hoc interpretability techniques, which often struggle with specificity and direct functional linkage.

While limitations regarding dataset scope, structure accuracy dependence, and scalability exist, ProteinRPN represents a significant step towards more accurate and insightful computational protein function annotation. In the next chapter, we conclude our findings and discuss the limitations and future scope of the model in detail.

Chapter 6

Conclusion and Future Work

Accurately predicting protein function is a key bottleneck in biological discovery, particularly useful in understanding disease mechanisms and guiding drug development. A key challenge in improving protein function prediction has been identifying specific structural regions or residues responsible for function, as many computational methods rely on post-hoc analyses with limited specificity. To address these challenges, this thesis introduced ProteinRPN, a novel deep learning framework designed for accurate protein function prediction with an emphasis on proactively identifying functionally relevant structural regions, using those to refine the protein representation learned by the model and finally predicting function.

ProteinRPN incorporates a graph-based Region Proposal Network (RPN), inspired by computer vision techniques, to first detect potential functional subgraph regions (anchors) within the protein structure graph. These proposals are then refined through a multi-stage process involving a hierarchy-aware Node Drop Pooling layer that integrates domain knowledge, such as secondary structure preference and spatial proximity of functional residues. A dedicated Functional Attention Layer further enhances the representations of predicted functional nodes while ensuring important contextual information from the broader graph structure is preserved. The model utilizes a Graph Multiset Transformer (GMT) to effectively aggregate node-level information into a comprehensive graph-level representation suitable for final prediction. The entire framework is optimized using a combination of standard cross-entropy loss and representation learning techniques, including supervised contrastive (SupCon) and self-supervised InfoNCE losses.

Comprehensive evaluations demonstrated that ProteinRPN significantly outperforms existing state-of-the-art methods, including sequence- and structure-based approaches like HEAL and

DeepFRI, across standard Protein Function Prediction benchmarks and metrics ($F_{\max}$, AUPR, $S_{\min}$) for all three Gene Ontology (GO) domains. Furthermore, qualitative visualizations confirmed ProteinRPN's ability to accurately localize known functional residues within protein structures, often identifying spatially clustered regions consistent with active or binding sites.

This work presents a robust and scalable deep learning approach that advances structure-based function prediction by directly integrating the identification of functionally important structural regions into the prediction pipeline. By providing predictions that are not only accurate but also grounded in specific structural regions, ProteinRPN offers improved interpretability and the potential for deeper biological insights compared to less localized methods.

6.1 Limitations of the Current Work

Despite its promising results, the current ProteinRPN framework has several limitations that suggest avenues for future improvement:

Dataset Scope: The evaluation was primarily conducted on established benchmarks derived from PDB and SWISS-MODEL; performance on broader datasets or species with different characteristics warrants further investigation. Furthermore, incorporating a more extensive annotated dataset at the pretraining stage could further enhance model performance

Structure Dependence: ProteinRPN inherently relies on the availability and quality of 3D structural information, and the impact of inaccuracies in predicted structures (e.g., from AlphaFold2) on region proposal and function prediction accuracy needs further characterization.

Specific Functional Classes: Performance variations across different GO domains (e.g., potentially lower relative performance on certain Biological Process (BP) categories compared to Molecular Function (MF) or Cellular Component (CC)) suggest that the model's effectiveness might differ depending on the nature of the function being predicted.

Interpretability Limits: Although the region proposal mechanism enhances interpretability, fully dissecting the complex interplay between the graph convolutions, attention scores,

and final predictions remains challenging for deep graph networks.

6.2 Future Work and Directions

Future work could focus on refining the region proposal algorithm to improve its biological relevance, perhaps incorporating learned priors or alternative anchor generation strategies. Several promising directions exist for extending and improving the ProteinRPN framework:

6.2.1 Integrating Multi-Modal Data

- Integrating sequence information more deeply, perhaps by fine-tuning Protein Language Models (PLMs) within a multi-modal ProteinRPN architecture or using joint sequence-structure embeddings, could capture richer biological context.
- Incorporating Protein-Protein Interaction (PPI) network information, potentially through graph fusion techniques or multi-graph GNNs, could specifically enhance the prediction of Biological Process (BP) terms [9].
- Exploring the integration of other data modalities, such as protein domain annotations (InterPro), textual information from literature, gene expression profiles, or post-translational modification (PTM) data, offers potential for more holistic functional understanding.
- Given the training set's inclusion of predicted structures using Alphafold2 [29], further characterization of performance using only predicted structures for both training and testing would be a valuable addition to assess the model's sensitivity to structural data source, particularly as the current architecture primarily utilizes contact maps rather than fine-grained atomic coordinates.

6.2.2 Addressing Broader Protein Function Prediction Challenges

- Adapting ProteinRPN or combining its region-focused approach with few-shot or zero-shot learning techniques could improve predictions for rare or novel GO terms.

- Explicitly modeling multi-functional proteins, potentially by allowing region proposals to map to distinct functional annotations, is an important direction.
- Further work could focus on enhancing the model's ability to discriminate between closely related functions that might share similar structural features.
- Applying transfer learning principles to adapt ProteinRPN, trained on well-annotated species, to predict functions in organisms with sparse experimental data is a valuable goal.

6.2.3 Downstream Applications

- The ability of ProteinRPN to pinpoint functional regions could significantly aid drug discovery by identifying potential binding sites or allosteric sites for therapeutic targeting.
- The predicted functionally relevant regions could serve as high-quality candidates for detailed experimental validation or computational docking studies to identify enzyme active sites or specific interaction interfaces.
- Insights from ProteinRPN's functional region identification could significantly streamline protein engineering workflows. By highlighting structurally localized areas critical for function, the model provides clear targets for site-directed mutagenesis aimed at tuning properties like substrate specificity or binding affinity. Conversely, it identifies regions potentially tolerant to modifications intended to improve stability or other biophysical characteristics without compromising the primary biological activity.

In conclusion, this thesis presented ProteinRPN, a novel structure-based protein function prediction method that leverages graph region proposals and hierarchy-aware attention to achieve improved predictive accuracy and enhanced localization of functional determinants. This work contributes to the rapidly advancing field of applying AI and deep learning to structural bioinformatics, demonstrating the potential of integrating concepts from other domains like computer vision and utilizing multi-stage refinement for complex biological problems.

Future development of ProteinRPN and similar region-aware, structure-based models holds significant promise for accelerating the annotation of the vast proteome, deepening our

understanding of the structure-function relationship, and ultimately aiding progress in medicine and biotechnology.

This page was intentionally left blank

Appendix A

Code listing

This appendix presents the Python implementation of the connected components loss function, as introduced and discussed in Chapter 3. The design prioritizes compatibility with gradient-based learning methods; all constituent operations are differentiable, thus ensuring that the model can be trained end-to-end with effective gradient flow through this loss component.

```
def weighted_degree(index, num_nodes, edge_weights):
    deg = torch.zeros(num_nodes, device=index.device)
    deg.scatter_add_(0, index, edge_weights)
    return deg

def connected_components_loss(node_scores, edge_index, batch, y_true,
    epsilon=1e-6, reg_lambda=0.01):
    num_graphs = batch.max().item() + 1
    loss_values = []

    for i in range(num_graphs):
        # Get nodes for current graph
        graph_mask = batch == i
        graph_scores = node_scores[graph_mask]

        # Subgraph extraction
        sub_edge_index, _ = subgraph(graph_mask, edge_index,
           relabel_nodes=True)
        num_nodes = graph_scores.size(0)

```

```
        edge_weights = graph_scores[sub_edge_index[0]] * graph_scores
            [sub_edge_index[1]]
        deg = weighted_degree(sub_edge_index[0], num_nodes,
            edge_weights)
        edge_weights_sum = edge_weights.sum()
        trace_L = torch.sum(deg) - edge_weights_sum

        # Normalize by expected number of functional residues
        norm_factor = torch.clamp(y_true[i], min=epsilon)
        loss_values.append(trace_L / norm_factor)

    connectivity_loss = torch.mean(torch.stack(loss_values))
    reg_term = reg_lambda * torch.mean(node_scores ** 2)
    total_loss = connectivity_loss + reg_term

    return total_loss, connectivity_loss, reg_term
```

References

[1] C. Perez-Iratxeta, G. Palidwor, and M. A. Andrade-Navarro. "Towards completion of the Earth's proteome". In: *EMBO Reports* 8.12 (2007), pp. 1135–1141. DOI: 10.1038/sj.embor.7401117.

[2] "UniProt: the universal protein knowledgebase in 2023". In: *Nucleic acids research* 51.D1 (2023), pp. D523–D531.

[3] J. Durairaj, A. M. Waterhouse, T. Mets, et al. "Uncovering new families and folds in the natural protein universe". In: *Nature* 622 (2023), pp. 646–653. DOI: 10.1038/s41586-023-06622-3.

[4] R. Aebersold, J. N. Agar, I. J. Amster, et al. "How many human proteoforms are there?" In: *Nature Chemical Biology* 14.3 (2018), pp. 206–214. DOI: 10.1038/nchembio.2576.

[5] A. LaPelusa and R. Kaushik. "Physiology, Proteins". In: *StatPearls*. Updated 2022 Nov 14. Treasure Island (FL): StatPearls Publishing, 2022.

[6] F. Boadu, A. Lee, and J. Cheng. "Deep learning methods for protein function prediction". In: *Proteomics* 25.1-2 (Jan. 2025). Epub 2024 Jul 12, e2300471. DOI: 10.1002/pmic.202300471.

[7] B. Alberts, A. Johnson, J. Lewis, et al. "The Shape and Structure of Proteins". In: *Molecular Biology of the Cell*. 4th. New York: Garland Science, 2002.

[8] B. Lin, X. Luo, Y. Liu, and X. Jin. "A comprehensive review and comparison of existing computational methods for protein function prediction". In: *Briefings in Bioinformatics* 25.4 (June 2024), bbae289. ISSN: 1477-4054. DOI: 10.1093/bib/bbae289. eprint: https://academic.oup.com/bib/article-pdf/25/4/bbae289/58286630/bbae289.pdf. URL: https://doi.org/10.1093/bib/bbae289.

[9] C. J. Jeffery. "Current successes and remaining challenges in protein function prediction". In: *Frontiers in Bioinformatics* Volume 3 - 2023 (2023). ISSN: 2673-7647. DOI: 10.3389/fbinf.2023.1222182. URL: https://www.frontiersin.org/journals/bioinformatics/articles/10.3389/fbinf.2023.1222182.

[10] J. A. Gerlt, P. C. Babbitt, and I. Rayment. "Divergent evolution in the enolase superfamily: The interplay of mechanism and specificity". In: *Archives of Biochemistry and Biophysics* 433 (2005), pp. 59–70. DOI: 10.1016/j.abb.2004.07.034.

[11] J. A. Gerlt, P. C. Babbitt, M. P. Jacobson, and S. C. Almo. "Divergent evolution in enolase superfamily: Strategies for assigning functions". In: *Journal of Biological Chemistry* 287 (2012), pp. 29–34. DOI: 10.1074/jbc.r111.240945.

[12] V. Gligorijević, P. D. Renfrew, T. Kosciolek, et al. "Structure-based protein function prediction using graph convolutional networks". In: *Nature Communications* 12 (2021), p. 3168. DOI: 10.1038/s41467-021-23303-9.

[13] Z. Gu, X. Luo, J. Chen, M. Deng, and L. Lai. "Hierarchical graph transformer with contrastive learning for protein function prediction". In: *Bioinformatics* 39.7 (June 2023), btad410. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btad410. eprint: https://academic.oup.com/bioinformatics/article-pdf/39/7/btad410/50874760/btad410.pdf. URL: https://doi.org/10.1093/bioinformatics/btad410.

[14] S. Ren, K. He, R. Girshick, and J. Sun. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. 2016. arXiv: 1506.01497 [cs.CV]. URL: https://arxiv.org/abs/1506.01497.

[15] V. Ivanisenko, D. Grigorovich, and N. Kolchanov. "PDBSite: a database on biologically active sites and their spatial surroundings in proteins with known tertiary structure". In: *The Second International Conference on Bioinformatics of Genome Regulation and Structure (BGRS'2000)*. Vol. 2. Novosibirsk, Russia, Aug. 2000, pp. 171–174.

[16] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. "The Protein Data Bank". In: *Nucleic Acids Research* 28 (2000), pp. 235–242. DOI: 10.1093/nar/28.1.235.

[17] G. Abraham, M. Hamilton, and R. van Schyndel. "Knowledge-Based Protein Structure Prediction using Approximate Occurrences". In.

[18] M. J. Lopez and S. S. Mohiuddin. "Biochemistry, Essential Amino Acids". In: *StatPearls*. Updated 2024 Apr 30. Treasure Island (FL): StatPearls Publishing, 2024.

[19] T. Umumararungu, N. Gahamanyi, J. Mukiza, et al. “Proline, a unique amino acid whose polymer, polyproline II helix, and its analogues are involved in many biological processes: a review”. In: *Amino Acids* 56.1 (2024). Published 2024 Aug 25, p. 50. DOI: 10.1007/s00726-024-03410-9.

[20] G. A. Petsko. “Dog eat dogma”. In: *Genome Biology* 1.2 (2000), comment1002.1–comment1002.2. DOI: 10.1186/gb-2000-1-2-comment1002.

[21] C. B. Anfinsen. “Principles that Govern the Folding of Protein Chains”. In: *Science* 181.4096 (1973), pp. 223–230. DOI: 10.1126/science.181.4096.223. URL: https://www.science.org/doi/10.1126/science.181.4096.223.

[22] F. Sanger and E. O. P. Thompson. “The amino-acid sequence in the glycyl chain of insulin. I. The identification of lower peptides from partial hydrolysates”. In: *Biochemical Journal* 53.3 (1953), pp. 353–366. DOI: 10.1042/bj0530353. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1198157/.

[23] D. L. Nelson. *Lehninger Principles of Biochemistry*. New York: W.H. Freeman, 2005.

[24] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. *Molecular Biology of the Cell*. 4th. New York: Garland Science, 2002. ISBN: 9780815332183. URL: https://www.ncbi.nlm.nih.gov/books/NBK21054/.

[25] L. Pauling, R. B. Corey, and H. R. Branson. “The structure of proteins; two hydrogen-bonded helical configurations of the polypeptide chain”. In: *Proceedings of the National Academy of Sciences of the United States of America* 37.4 (1951), pp. 205–211. DOI: 10.1073/pnas.37.4.205. URL: https://www.pnas.org/doi/10.1073/pnas.37.4.205.

[26] A. Perczel, Z. Gáspári, and I. G. Csizmadia. “Structure and stability of beta-pleated sheets”. In: *Journal of Computational Chemistry* 26.11 (2005), pp. 1155–1168. DOI: 10.1002/jcc.20255.

[27] I. Rehman, C. C. Kerndt, and S. Botelho. “Biochemistry, Tertiary Protein Structure”. In: *StatPearls [Internet]*. [Updated 2024 May 25]. Treasure Island (FL): StatPearls Publishing, 2025. URL: https://www.ncbi.nlm.nih.gov/books/NBK470269/.

[28] M. Rashid, F. Khatib, and A. Sattar. “Protein preliminaries and structure prediction fundamentals for computer scientists”. In: *http://arxiv.org* (Oct. 2015).

[29] J. Jumper, R. Evans, A. Pritzel, et al. “Highly accurate protein structure prediction with AlphaFold”. In: *Nature* 596 (2021), pp. 583–589. DOI: 10.1038/s41586-021-03819-2.

[30] “The Gene Ontology resource: enriching a GOld mine”. In: *Nucleic acids research* 49.D1 (2021), pp. D325–D334.

[31] S. A. Aleksander, J. Balhoff, S. Carbon, J. M. Cherry, H. J. Drabkin, D. Ebert, M. Feuermann, P. Gaudet, N. L. Harris, et al. "The gene ontology knowledgebase in 2023". In: *Genetics* 224.1 (2023), iyad031.

[32] G. Valentini. "True path rule hierarchical ensembles for genome-wide gene function prediction". In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 8.3 (2011), pp. 832–847. DOI: 10.1109/TCBB.2010.38.

[33] P. Radivojac, W. Clark, T. Oron, et al. "A large-scale evaluation of computational protein function prediction". In: *Nature Methods* 10 (2013), pp. 221–227. DOI: 10.1038/nmeth.2340.

[34] Z. Gu, X. Luo, J. Chen, M. Deng, and L. Lai. "Hierarchical graph transformer with contrastive learning for protein function prediction". In: *Bioinformatics* 39.7 (2023), btad410.

[35] K. O'Shea and R. Nash. *An Introduction to Convolutional Neural Networks*. 2015. arXiv: 1511.08458 [cs.NE]. URL: https://arxiv.org/abs/1511.08458.

[36] T. N. Kipf and M. Welling. *Semi-Supervised Classification with Graph Convolutional Networks*. 2017. arXiv: 1609.02907 [cs.LG]. URL: https://arxiv.org/abs/1609.02907.

[37] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. *Graph Attention Networks*. 2018. arXiv: 1710.10903 [stat.ML]. URL: https://arxiv.org/abs/1710.10903.

[38] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762.

[39] A. Rives et al. "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences". In: *Proceedings of the National Academy of Sciences* 118.15 (2021), e2016239118. DOI: 10.1073/pnas.2016239118. URL: https://www.pnas.org/doi/10.1073/pnas.2016239118.

[40] A. Elnaggar et al. "ProtTrans: Toward Understanding the Language of Life Through Self-Supervised Learning". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.10 (2022), pp. 7112–7127. DOI: 10.1109/TPAMI.2021.3095381. URL: https://doi.org/10.1109/TPAMI.2021.3095381.

[41] J. Su, C. Han, Y. Zhou, J. Shan, X. Zhou, and F. Yuan. "SaProt: Protein Language Modeling with Structure-aware Vocabulary". In: *bioRxiv* (2023). DOI: 10.1101/2023.10.01.560349. eprint: https://www.biorxiv.org/content/early/2023/10/02/2023.10.01.560349.1.full.pdf. URL: https://www.biorxiv.org/content/early/2023/10/02/2023.10.01.560349.1.

[42] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan. *Supervised Contrastive Learning*. 2021. arXiv: 2004.11362 [cs.LG]. URL: https://arxiv.org/abs/2004.11362.

[43] A. van den Oord, Y. Li, and O. Vinyals. *Representation Learning with Contrastive Predictive Coding*. 2019. arXiv: 1807.03748 [cs.LG]. URL: https://arxiv.org/abs/1807.03748.

[44] V. T. D. Nguyen and T.-S. Hy. *Advances in Protein Representation Learning: Methods, Applications, and Future Directions*. 2025. arXiv: 2503.16659 [cs.LG]. URL: https://arxiv.org/abs/2503.16659.

[45] A. Madani, B. McCann, N. Naik, N. S. Keskar, N. Anand, R. R. Eguchi, P.-S. Huang, and R. Socher. *ProGen: Language Modeling for Protein Generation*. 2020. arXiv: 2004.03497 [q-bio.BM]. URL: https://arxiv.org/abs/2004.03497.

[46] X. Wang, Z. Zheng, F. Ye, D. Xue, S. Huang, and Q. Gu. *DPLM-2: A Multimodal Diffusion Protein Language Model*. 2024. arXiv: 2410.13782 [cs.LG]. URL: https://arxiv.org/abs/2410.13782.

[47] V. G. Satorras, E. Hoogeboom, and M. Welling. *E(n) Equivariant Graph Neural Networks*. 2022. arXiv: 2102.09844 [cs.LG]. URL: https://arxiv.org/abs/2102.09844.

[48] B. Jing, S. Eismann, P. Suriana, R. J. L. Townshend, and R. Dror. "Learning from Protein Structure with Geometric Vector Perceptrons". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=1YLJDvSx6J4.

[49] Z. Zhang, M. Xu, A. R. Jamasb, V. Chenthamarakshan, A. Lozano, P. Das, and J. Tang. "Protein Representation Learning by Geometric Structure Pretraining". In: *The Eleventh International Conference on Learning Representations*. 2023. URL: https://openreview.net/forum?id=to3qCB3tOh9.

[50] T. Hayes et al. "Simulating 500 million years of evolution with a language model". In: *bioRxiv* (2024). DOI: 10.1101/2024.07.01.600583. eprint: https://www.biorxiv.org/content/early/2024/07/02/2024.07.01.600583.full.pdf. URL: https://www.biorxiv.org/content/early/2024/07/02/2024.07.01.600583.

[51] S. Iqbal, F. Ge, F. Li, T. Akutsu, Y. Zheng, R. B. Gasser, D. J. Yu, G. I. Webb, and J. Song. "PROST: AlphaFold2-aware Sequence-Based Predictor to Estimate Protein Stability Changes upon Missense Mutations". In: *Journal of Chemical Information and Modeling* 62.17 (2022), pp. 4270–4282. DOI: 10.1021/acs.jcim.2c00799. URL: https://doi.org/10.1021/acs.jcim.2c00799.

[52] M. Baek et al. "Accurate prediction of protein structures and interactions using a three-track neural network". In: *Science* 373.6557 (2021), pp. 871–876. DOI: 10.1126/science.abj8754. URL: https://www.science.org/doi/10.1126/science.abj8754.

[53] R. Wu et al. "High-resolution de novo structure prediction from primary sequence". In: *bioRxiv* (2022). DOI: 10.1101/2022.07.21.500999. eprint: https://www.biorxiv.org/content/early/2022/07/22/2022.07.21.500999.full.pdf. URL: https://www.biorxiv.org/content/early/2022/07/22/2022.07.21.500999.

[54] J.-Y. Chen, J.-F. Wang, Y. Hu, X.-H. Li, Y.-R. Qian, and C.-L. Song. "Evaluating the advancements in protein language models for encoding strategies in protein function prediction: a comprehensive review". In: *Frontiers in Bioengineering and Biotechnology* 13 (2025), p. 1506508. DOI: 10.3389/fbioe.2025.1506508.

[55] M. Kulmanov, M. A. Khan, and R. Hoehndorf. "DeepGO: predicting protein functions from sequence and interactions using a deep ontology-aware classifier". In: *Bioinformatics* 34.4 (Oct. 2017), pp. 660–668. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btx624. eprint: https://academic.oup.com/bioinformatics/article-pdf/34/4/660/48914540/bioinformatics_34_4_660.pdf. URL: https://doi.org/10.1093/bioinformatics/btx624.

[56] M. Kulmanov and R. Hoehndorf. "DeepGOPlus: improved protein function prediction from sequence". In: *Bioinformatics* 36.2 (July 2019), pp. 422–429. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btz595. eprint: https://academic.oup.com/bioinformatics/article-pdf/36/2/422/48990969/btz595.pdf. URL: https://doi.org/10.1093/bioinformatics/btz595.

[57] J. Wu, H. Qing, J. Ouyang, J. Zhou, Z. Gao, C. E. Mason, Z. Liu, and T. Shi. "HiFun: homology independent protein function prediction by a novel protein-language self-attention model". In: *Briefings in Bioinformatics* 24.5 (Aug. 2023), bbad311. ISSN: 1477-4054. DOI: 10.1093/bib/bbad311. eprint: https://academic.oup.com/bib/article-pdf/24/5/bbad311/51711254/bbad311.pdf. URL: https://doi.org/10.1093/bib/bbad311.

[58] Y. Cao and Y. Shen. "TALE: Transformer-based protein function Annotation with joint sequence–Label Embedding". In: *Bioinformatics* 37.18 (Mar. 2021), pp. 2825–2833. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btab198. eprint: https://academic.oup.com/bioinformatics/article-pdf/37/18/2825/50579661/btab198.pdf. URL: https://doi.org/10.1093/bioinformatics/btab198.

[59] P. Jiao, B. Wang, X. Wang, B. Liu, Y. Wang, and J. Li. "Struct2GO: protein function prediction based on graph pooling algorithm and AlphaFold2 structure information". In: *Bioinformatics* 39.10 (Oct. 2023), btad637. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btad637. eprint: https://academic.oup.com/bioinformatics/article-pdf/39/10/btad637/52673512/btad637.pdf. URL: https://doi.org/10.1093/bioinformatics/btad637.

[60] R. Sharan, I. Ulitsky, and R. Shamir. "Network-based prediction of protein function". In: *Molecular Systems Biology* 3 (2007), p. 88. DOI: 10.1038/msb4100129. URL: https://www.embopress.org/doi/10.1038/msb4100129.

[61] D. Warde-Farley, S. L. Donaldson, O. Comes, et al. "The GeneMANIA prediction server: biological network integration for gene prioritization and predicting gene function". In: *Nucleic Acids Research* 38.Web Server issue (2010), W214–W220. DOI: 10.1093/nar/gkq537.

[62] V. Gligorijević, M. Barot, and R. Bonneau. "deepNF: deep network fusion for protein function prediction". In: *Bioinformatics* 34.22 (2018), pp. 3873–3881. DOI: 10.1093/bioinformatics/bty440.

[63] M. Barot, V. Gligorijević, K. Cho, and R. Bonneau. "NetQuilt: deep multispecies network-based protein function prediction using homology-informed network similarity". In: *Bioinformatics* 37.16 (2021), pp. 2414–2422. DOI: 10.1093/bioinformatics/btab098.

[64] A. Rives, J. Meier, T. Sercu, S. Goyal, Z. Lin, J. Liu, D. Guo, M. Ott, C. L. Zitnick, J. Ma, et al. "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences". In: *Proceedings of the National Academy of Sciences* 118.15 (2021), e2016239118.

[65] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. "The Protein Data Bank". In: *Nucleic Acids Research* 28.1 (2000), pp. 235–242. DOI: 10.1093/nar/28.1.235.

[66] T. N. Kipf and M. Welling. *Semi-Supervised Classification with Graph Convolutional Networks*. 2017. arXiv: 1609.02907 [cs.LG]. URL: https://arxiv.org/abs/1609.02907.

[67] G. J. Bartlett, C. T. Porter, N. Borkakoti, and J. M. Thornton. "Analysis of Catalytic Residues in Enzyme Active Sites". In: *Journal of Molecular Biology* 324.1 (2002), pp. 105–121. ISSN: 0022-2836. DOI: https://doi.org/10.1016/S0022-2836(02)01036-7. URL: https://www.sciencedirect.com/science/article/pii/S0022283602010367.

[68] W. G. Touw, C. Baakman, J. Black, T. A. H. te Beek, E. Krieger, R. P. Joosten, and G. Vriend. "A series of PDB related databases for everyday needs". In: *Nucleic Acids Research* 43.Database issue (Jan. 2015), pp. D364–D368. DOI: 10.1093/nar/gku1028.

[69] J. Baek, M. Kang, and S. J. Hwang. *Accurate Learning of Graph Representations with Graph Multiset Pooling*. 2021. arXiv: 2102.11533 [cs.LG]. URL: https://arxiv.org/abs/2102.11533.

[70] J. M. Dana, A. Gutmanas, N. Tyagi, G. Qi, C. O'Donovan, M. Martin, and S. Velankar. "SIFTS: updated Structure Integration with Function, Taxonomy and Sequences resource allows 40-fold increase in coverage of structure-based annotations for proteins". In: *Nucleic Acids Research* 47.D1 (Nov. 2018), pp. D482–D489. ISSN: 0305-1048. DOI: 10.1093/nar/gky1114. eprint: https://academic.oup.com/nar/article-pdf/47/D1/D482/27437256/gky1114.pdf. URL: https://doi.org/10.1093/nar/gky1114.

[71] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. "Basic local alignment search tool". In: *Journal of Molecular Biology* 215.3 (1990), pp. 403–410. DOI: 10.1016/S0022-2836(05)80360-2.

[72] S. Das, D. Lee, I. Sillitoe, N. L. Dawson, J. G. Lees, and C. A. Orengo. "Functional classification of CATH superfamilies: a domain-based approach for protein function annotation". In: *Bioinformatics* 31.21 (July 2015), pp. 3460–3467.

[73] L. Meng and X. Wang. "TAWFN: a deep learning framework for protein function prediction". In: *Bioinformatics* 40.10 (Sept. 2024), btae571. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btae571. eprint: https://academic.oup.com/bioinformatics/article-pdf/40/10/btae571/59491578/btae571.pdf. URL: https://doi.org/10.1093/bioinformatics/btae571.

[74] M. Kulmanov, F. J. Guzmán-Vega, P. D. Roggli, L. Lane, S. T. Arold, and R. Hoehndorf. "DeepGO-SE: Protein function prediction as Approximate Semantic Entailment". In: *bioRxiv* (2023). DOI: 10.1101/2023.09.26.559473. eprint: https://www.biorxiv.org/content/early/2023/09/28/2023.09.26.559473.full.pdf. URL: https://www.biorxiv.org/content/early/2023/09/28/2023.09.26.559473.

[75] C. Deng, Z. Yue, and Z. Zhang. "Polynormer: Polynomial-Expressive Graph Transformer in Linear Time". In: *The Twelfth International Conference on Learning Representations*. 2024. URL: https://openreview.net/forum?id=hmv1LpNfXa.

[76] P. Feeney and M. C. Hughes. *SINCERE: Supervised Information Noise-Contrastive Estimation REvisited*. 2024. arXiv: 2309.14277 [cs.CV]. URL: https://arxiv.org/abs/2309.14277.

[77] T. Pan, C. Li, Y. Bi, Z. Wang, R. B. Gasser, A. W. Purcell, T. Akutsu, G. I. Webb, S. Imoto, and J. Song. "PFresGO: an attention mechanism-based deep-learning approach for protein annotation by integrating gene ontology inter-relationships". In: *Bioinformatics* 39.3 (2023), btad094.