Antigen-aware Deep Learning Model for Antibody Sequence Generation.

By

Muhammad Inam Rafique

(Registration No: 00000402183)

Department of Sciences

School of Interdisciplinary Engineering & Sciences (SINES)

National University of Sciences & Technology (NUST)

Islamabad, Pakistan

(2024)

Antigen-aware Deep Learning Model for Antibody Sequence Generation.

By

Muhammad Inam Rafique

(Registration No: 00000402183)

A thesis submitted to the National University of Sciences & Technology, Islamabad,

In partial fulfilment of the requirements for the degree of

Master of Science in
Bioinformatics

Supervisor: Dr. Mehak Rafiq

School of Interdisciplinary Engineering & Sciences (SINES)

National University of Sciences & Technology (NUST)

Islamabad, Pakistan

(2024)

Annex A to NUST Letter No. 0972/102/Exams/Thesis-Cert dated 23 Dec 16.

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS/MPhil thesis written by Mr. Muhammad Inam Rafique Registration No. 00000402183 of SINES has been vetted by the undersigned, found complete in all aspects as per NUST Statutes/Regulations, is free of plagiarism, errors, and mistakes and is accepted as partial fulfillment for award of MS/MPhil degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in the said thesis.

Signature with stamp: DR. MEHAK RAFIQ Assistant Professor SINES, National University of Science & Technology Islamabad

Name of Supervisor: Dr Mehak Rafiq

Date: 12-09-2024

Signature of HoD with stamp: Dr. Fouzia Malik HoD Sciences Professor SINES NUST Sector H 12 Islamabad

Date: 13-9-2024

Countersign by

Signature (Dean/Principal):

Date: 13/09/2024

AUTHOR'S DECLARATION

I Muhammad Inam Rafique hereby state that my MS thesis titled **Antigen-aware Deep Learning Model for Antibody Sequence Generation** is my work and has not been submitted previously by me for taking any degree from the National University of Sciences and Technology, Islamabad, or anywhere else in the country/ world.

At any time if my statement is found to be incorrect even after I graduate, the university has the right to withdraw my MS degree.

Name of Student: Muhammad Inam Rafique

Date: September 18, 2024

PLAGIARISM UNDERTAKING

I solemnly declare that the research work presented in the thesis titled **"Antigen-aware Deep Learning Model for Antibody Sequence Generation"** is solely my research work with no significant contribution from any other person. Small contribution/ help wherever taken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero-tolerance policy of the HEC and the National University of Sciences and Technology (NUST), Islamabad towards plagiarism. Therefore, I as an author of the above-titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred to/cited.

I undertake that if I am found guilty of any formal plagiarism in the above-titled thesis even after the award of the MS degree, the University reserves the right to withdraw/revoke my MS degree and that HEC and NUST, Islamabad has the right to publish my name on the HEC/University website on which names of students are placed who submitted plagiarized thesis.

Name: Muhammad Inam Rafique

DEDICATION

I dedicate this thesis to my beloved parents and siblings, whose prayers, love, encouragement, and endless support made my success possible, and to my teachers, whose help and guidance helped me throughout my academic journey. I also want to dedicate my work to my Late uncle, Muhammad Hanif who always encouraged me and was my strength.

ACKNOWLEDGMENT

First, I am deeply grateful to **Allah Almighty,** whose countless blessings and guidance have brought me to this point in my educational journey. None of this would have been possible without His grace and mercy. I would like to express my deepest gratitude to my **Parents,** whose unwavering support and endless sacrifices have been my pillars of strength. Their encouragement and belief in me have always been the driving force behind my achievements.

I want to express my sincere gratitude to my supervisor **Dr. Mehak Rafiq.** Your guidance, patience, and invaluable guidance have been of great help in completing this thesis. Your encouragement and belief in my abilities have helped me through the challenges of this research.

I am deeply thankful to my GEC members, **Dr. Ishrat Jabeen, Dr. Shahzad Younis,** and **Dr. Masood Ur Rehman Kayani,** for their insightful guidance and constructive feedback. Your expertise and advice have helped shape this thesis and guided me through the research process. I am extremely grateful to my teachers **Mr. Osaf Shah** and **Mr. & Mrs. Irfan Naveed** for their everlasting support, encouragement and prayers.

I would like to extend my gratitude to my friends **Syed Usman, Zeeshan Waqar, Zill-e-Noor**, and **Maryam Areej** for their continuous support throughout this journey. A special thanks to **Ayesha Sadiqa** for her unwavering assistance at every stage. I am also thankful to my colleagues from the Data Analytics Lab, particularly **Amman Safeer**, for their valuable collaboration and encouragement during the course of my research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Advancements in artificial intelligence over the last decade have transformed numerous fields, including biotechnology. Recent developments in deep learning (DL) have led to the creation of models capable of generating antibody sequences with remarkable efficiency. These models, built on cutting-edge NLP-based architectures and trained on extensive datasets of protein sequences, harness the inherent information encoded in protein sequences, from structural conformations to binding affinities. By leveraging deep learning, these methods can potentially reduce the reliance on traditional, resource-intensive experimental procedures for antibody development. However, antigen-specific antibody sequence generation is still a problem that needs to be addressed. In this study, AbAtT5, a finetuned transformer-based model specifically designed for generating antigen-specific antibodies using full-length antigen sequences is introduced. AbAtT5 is finetuned on a large protein language model, protT5, harnessing the potential of transfer learning by updating the weights and biases of the pre-trained model. AbAtT5 demonstrated superior performance compared to existing models like HERN and EAGLE, achieving improvements of up to 1.88% in VAAR and 18.04% in SeqID. These findings underscore the model's potential to accelerate the antibody design process by providing more accurate sequence generation. The ability of AbAtT5 to generate antibodies with higher sequence identity and alignment rates highlights its promise as a powerful tool in the field of computational antibody generation, offering a more efficient approach to identifying potent antigen-specific antibody candidates.

CHAPTER 1: INTRODUCTION

This chapter discusses foundational concepts relevant to this study, including proteins, antibodies, and the role of computational methods in antibody engineering. It also discusses the emerging applications of generative AI and deep learning in biology, leading to the identification of the problem statement and proposed solution

1.1. Proteins

Proteins are fundamental macromolecules that perform many functions within biological systems. These proteins are made up of amino acids joined together via peptide bonds. The linear combination of amino acids determines the structure and function of proteins and is based on which proteins are classified into different families. The sequence of proteins is known as primary structure forming chains, these chains fold into more complex forms, contributing to the formation of secondary, tertiary, and quaternary structures [1] Proteins carry out all the functions in a living body, encompassing enzymatic catalysis, providing structural support, facilitating signaling pathways, and mediating immune responses [1] [2], Figure1.1.

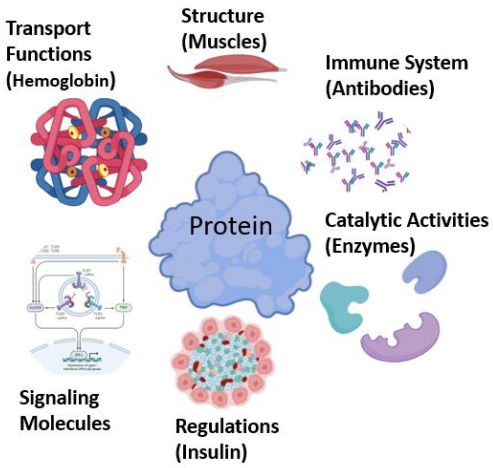

Figure 1.1 Various Functions of Proteins in the Body.
Proteins perform several vital roles, including transport functions (Hemoglobin), structural support (Muscles), immune response (Antibodies), catalytic activities (Enzymes), signaling molecules, and regulation (Insulin)

.

As mentioned earlier, the secondary structure arises from the primary structure, which refers to local conformations such as alpha-helices and beta-sheets stabilized by hydrogen bonds. These structures further fold into the tertiary structure, representing

the overall 3D shape of a single polypeptide chain, stabilized by various interactions including hydrogen bonds, and ionic and hydrophobic interactions [3]. In some proteins, multiple polypeptide chains (subunits) come together to form the quaternary structure, which is critical for the protein's function. This transition of protein from simple primary structure to quaternary structure is illustrated in Figure 1.2 below.

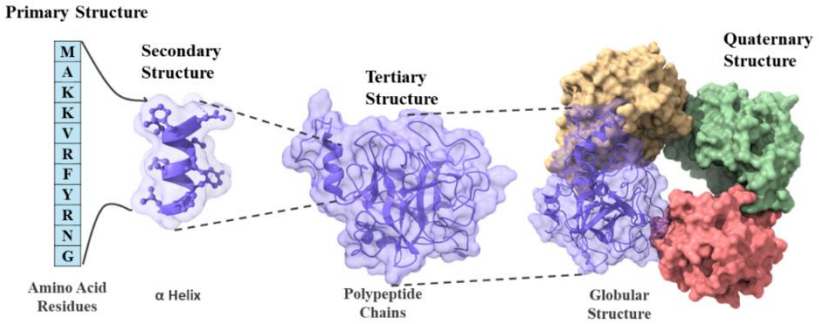

Figure 1.2 Transition of proteins from primary to quaternary structure.

The image illustrates the hierarchical structure of proteins, from the primary amino acid sequence to the complex quaternary structure formed by multiple polypeptide chains. Each level contributes to the protein's overall shape and function.

Variations in the amino acid sequence can lead to alterations in the protein's structure and function. Such changes can have significant implications, ranging from loss of function to the development of diseases [4]. For instance, a single amino acid substitution, valine replaces glutamic acid, causes sickle cell anemia, and hemoglobin loses the ability to carry oxygen [5]. On the other hand, these mutations can lead to possible interactions between the antibodies and antigens, which will be discussed below in detail. Thus, understanding these relationships between sequence, structure, and function is crucial. In the below section, a special class of proteins called antibodies is discussed which is the scope of this study.

1.2. Antibodies

Antibodies are immunoglobulin proteins produced by the immune system; they serve as the first line of defense against the attack of foreign particles known as antigens [6]. Antibodies are produced by B cells of the immune system; these B cells produce antibodies at a very high rate, approximately 2000 molecules per second, thus

generating an army of antibody molecules to fight against the invasion quickly [7]. Further details regarding these vital molecules are discussed in the following section.

1.2.1. Sequence and Structure of Antibodies

An antibody molecule typically possesses a Y-shaped structure in a globular form as shown in Figure1.3 (left), each antibody is made up of four polypeptide chains, two identical heavy chains (H) and two identical light chains (L), connected by disulfide bonds. The Y-shaped structure has two main regions: the Fab (fragment antigen-binding) region and the Fc (fragment crystallizable) region also shown in Figure 1.3 (Right) [8].

- **Fab Region**: This includes the variable regions of the heavy and light chains. The tips of the Y-shaped structure contain the antigen-binding sites, which are highly variable and allow for the specific recognition of antigens. Each Fab region can bind to an antigen, and the specificity is determined by the unique amino acid sequence in the variable region.
- **Fc Region**: This is the constant region that interacts with cell surface receptors (Fc receptors) and other immune molecules like complement proteins. The Fc region determines the effector function of the antibody, such as recruiting other immune cells or activating the complement system.

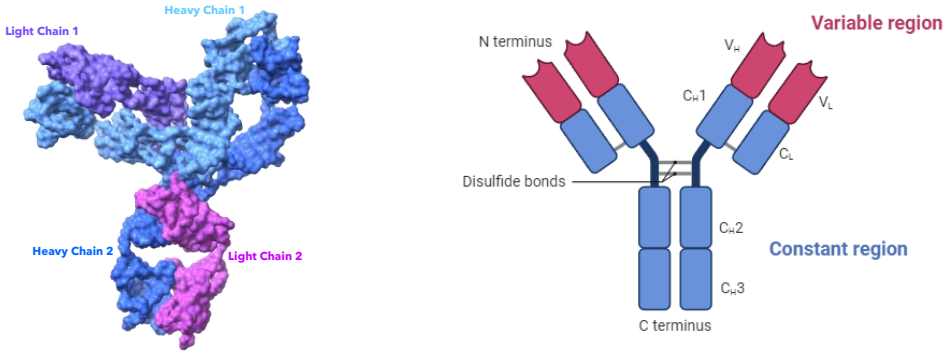

Figure 1.3 The Y-shaped structure of an antibody.

3D structure of an antibody (left) with two heavy chains (blue) and two light chains (purple). The schematic (right) highlights the variable region (red) for antigen binding and the constant region (blue) for immune function, connected by disulfide bonds

Regarding the sequential arrangement of the chains, each chain comprises four framework regions (FWRs) and three complementarity-determining regions (CDRs).

These CDRs are loops by structure and are part of Fab regions where the antigen binds and these are variable in lengths whereas CDR3 of heavy chain is the most important and hypervariable region, and most of the interactions between the antigen and antibody occur at this part of the antibody. This variability in CDR3 of the heavy chain allows antibodies to recognize a vast array of antigens. Talking about the length of the chains, a single light chain in mammals consists of about 220 amino acids while a heavy chain consists of 440 amino acids, however, in most research only the variable domains of both chains are considered with the approximate lengths of 110 and 120 for light chain and heavy chain respectively. In these chains, the FWRs and CDRs are determined using different numbering schemes [9]. These numbering schemes are systematic methods used to label and align the amino acid positions within the variable regions of antibody sequences. These schemes provide a standardized way to identify equivalent residues across different antibodies, facilitating comparison, analysis, and structural modeling. These numbering schemes are discussed below.

1.2.2. Antibody Numbering Schemes

Antibody numbering schemes are developed by different research institutes over the period. Antibody numbering schemes help in the identification and annotation of functional regions, such as CDRs, and are helpful in engineering and optimizing antibodies for therapeutic and diagnostic purposes. The most common schemes are as follows.

- **Kabat Numbering Scheme:**

Developed by Elvin Kabat, this scheme is based on the variability of amino acids in the sequences of antibodies. It assigns numbers to residues in the variable regions of both heavy and light chains, considering insertions and deletions observed in different antibodies. This scheme is widely used for studying antibody diversity and structure-function relationships [10].

- **Chothia Numbering Scheme:**

Introduced by Chothia and Lesk, this scheme is based on the structural features of antibodies. It aligns sequences according to the positions of the residues in the three-dimensional structure of antibodies. The Chothia scheme focuses on the

complementarity-determining regions (CDRs) and the framework regions, facilitating the study of antibody-antigen interactions [11].

- **IMGT Numbering Scheme:**

The International ImMunoGeneTics (IMGT) information system developed this scheme to provide a standardized and comprehensive framework for numbering immunoglobulin and T-cell receptor variable regions. The IMGT scheme aligns sequences according to their structural and functional properties, ensuring consistency across different antibodies and species [12].

These schemes allow researchers and developers to analyze, compare, and optimize antibodies for several applications effectively, ensuring uniformity and reliability in antibody research and development. This study follows the IMGT scheme because the data from sabdab is renumbered according to this numbering scheme.

1.2.3. Applications of Antibodies [13]

As of now, the structure of antibodies has been discussed; in this section, the applications of antibodies are briefly mentioned. Naturally, antibodies protect the body against malfunctioning proteins within the body as well as foreign invading pathogens. Knowing their specificity and efficiency to bind with antigens, many antibodies were developed and used for different purposes along with therapeutic purposes. They have a vast range of applications in treating different conditions, diagnosis, and research. Some of the applications in the said domains are briefly mentioned below but are not limited to these.

Therapeutics

Monoclonal Antibodies (mAbs) are engineered to target specific antigens, such as those found on cancer cells, pathogens, or involved in autoimmune diseases. They can function by directly neutralizing a pathogen, blocking a receptor involved in disease progression, or marking a diseased cell for destruction by the immune system. For example, Trastuzumab (Herceptin) is a monoclonal antibody used to treat HER2-positive breast cancer. It targets the HER2 receptor, a protein overexpressed in some breast cancer cells, inhibiting cell growth and survival. Antibody-drug Conjugates

(ADCs), are antibodies linked to a drug or toxic agent, allowing the specific delivery of the toxic agent to the targeted cells, minimizing side effects on healthy cells. Ado-trastuzumab emtansine (Kadcyla) is an ADC where trastuzumab is linked to a chemotherapy drug, used to target and kill HER2-positive breast cancer cells [14].

Diagnostics

Immunohistochemistry (IHC), IHC involves staining tissues with antibodies to detect the presence of specific markers. This technique is widely used in diagnosing diseases and determining the molecular characteristics of tumors. PD-L1 staining in tumor samples helps to identify patients who may benefit from PD-L1 inhibitor therapies in oncology. Enzyme-linked Immunosorbent Assay (ELISA), utilizes antibodies to detect the presence of antigens in a sample, commonly used for diagnosing infections and conditions. ELISA tests for HIV detection measure antibodies against HIV in a patient's blood, providing a reliable diagnostic method [15].

Research

In Flow Cytometry, Antibodies conjugated with fluorescent molecules are used to label specific cell populations. This method is fundamental in research for cell sorting, biomarker detection, and immune profiling. Antibodies against CD4 and CD8 are used to identify and quantify T helper cells and cytotoxic T cells in immunological studies [16].

The same is true with Western Blotting, in this technique antibodies are used to detect specific proteins in a sample, helping in studying protein expression, size, and modification. Antibodies against phosphorylated proteins can help in studying the activation states of various signaling pathways involved in cancer or other diseases. Immunoprecipitation, this technique uses antibodies to precipitate a specific antigen out of solution, allowing for the study of protein interactions and function. Immunoprecipitation of an oncogene like MYC tagged with an epitope can help to identify interacting partners and downstream effects in cellular pathways.

Each application of antibodies leverages their ability to bind specifically to their target antigen, highlighting their versatility and critical role in advancing medical

science, therapeutic interventions, and our understanding of complex biological systems.

1.2.4. Antibody Development: Methods & Limitations

In the above sections, a brief discussion of the applications of antibodies in different domains, these antibodies are developed and optimized for their specific use in those domains. Antibody development aims to create antibodies with improved specificity, affinity, and stability for therapeutic applications and efficient research tools [14]. Two types of approaches are in practice for antibody engineering, Rational approach and Empirical methods. In a rational approach, the structural knowledge of antibodies acquired from techniques like NMR and X-ray crystallography is used to develop and optimize antibodies of interest by generating a few number of potential candidates. In the empirical approach, large numbers of potential candidates are generated using methods such as hybridoma technology and phage display, and then different screening approaches are used to get the potential candidates [17]. Hybridoma technology involves immunizing an animal, typically a mouse, with an antigen to stimulate an immune response. B-cells producing the desired antibody are then harvested and fused with myeloma cells to create hybridomas. These hybridomas can be cultured indefinitely, producing monoclonal antibodies that are specific to the antigen used for immunization. This method has been foundational in the development of many therapeutic antibodies currently in use [18].

Phage display involves expressing antibody fragments on the surface of bacteriophages. This allows for the selection of antibodies that bind to a specific antigen by exposing the phage library to the antigen and selecting those that bind with high affinity. The selected phages are then amplified, and the process is repeated to enrich for high-affinity binders. Phage display has been widely used to develop antibodies that are difficult to obtain through traditional immunization methods [19].

1.2.5. Limitations of Traditional Antibody Development Methods

However, these traditional methods are often time-consuming and labor-intensive. Hybridoma technology, for example, involves several stages: immunizing an animal, creating hybridomas, screening for the desired antibody, and scaling up

production. This entire process can take several months to complete. Similarly, phage display requires multiple rounds of selection and amplification to isolate high-affinity binders, making it a lengthy process as well. These methods, while effective, are not always efficient, leading to a demand for faster, more automated approaches [20].

1.2.6. Computational Methods in Antibody Engineering

In recent years, computational methods have emerged as powerful tools in antibody engineering. These methods leverage bioinformatics and structural biology to predict and design antibody sequences with desired properties. Computational approaches can analyze large datasets to identify patterns and make predictions, significantly accelerating the antibody development process. Techniques such as molecular docking, molecular dynamics simulations, and sequence alignment are commonly used to predict antibody-antigen interactions and optimize antibody sequences [20].

1.3. Generative AI

Generative AI is a fascinating and rapidly evolving subset of artificial intelligence focused on creating new content, such as images, text, music, and even videos. Unlike traditional AI, which typically analyzes existing data to make predictions or classifications, generative AI models are designed to generate data that mimics the patterns and characteristics of the training data. The core of generative AI lies in its ability to learn from vast amounts of data and then use that knowledge to produce new, original content. This process often involves complex algorithms and neural networks, with some of the most prominent approaches being Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), and Transformers [21].

Generative Adversarial Networks, introduced by Ian Goodfellow and his colleagues in 2014, consist of two neural networks: a generator and a discriminator. The generator creates new data instances, while the discriminator evaluates them. The two networks are trained simultaneously in a process where the generator aims to produce increasingly realistic data, and the discriminator attempts to distinguish between real and generated data. This adversarial setup results in highly realistic

outputs, making GANs particularly effective in creating images, videos, and even deepfake content [22].

Variational Autoencoders, on the other hand, are a type of autoencoder that imposes a probabilistic structure on the latent space of the data. VAEs are particularly useful for generating new data points that are like the original dataset but introduce slight variations. This makes them valuable for tasks such as image synthesis, anomaly detection, and data compression.

Transformers represent a groundbreaking architecture in the realm of generative AI, especially for text generation. Introduced in the paper "Attention is All You Need [23]" by Vaswani et al. in 2017, Transformers have revolutionized natural language processing (NLP) with their ability to handle sequential data more effectively than previous models like recurrent neural networks (RNNs) and long short-term memory networks (LSTMs). The key innovation of Transformers is the self-attention mechanism, which allows the model to weigh the importance of different words in a sentence relative to each other, capturing context more accurately [24]. One of the most notable advancements using Transformer architecture is the development of large language models, such as OpenAI's GPT-3 and GPT-4. These models are capable of understanding and generating human-like text based on the context provided. They have been used in various applications, from chatbots and virtual assistants to content creation and code generation. The scalability of Transformers has enabled these models to achieve remarkable performance by training on diverse and extensive datasets [24].

Generative AI represents a significant leap forward in the field of artificial intelligence, offering new possibilities for creativity and innovation. The introduction of Transformer models has further amplified its potential, particularly in natural language processing. By understanding and addressing the challenges associated with generative AI, one can harness its potential to drive progress across various domains while ensuring ethical and responsible use.

1.3.1. Framework of Generative AI

Generative AI in Natural Language Processing (NLP) involves creating models that can generate human-like text based on given inputs. The process of developing a

generative AI model for NLP typically involves several key steps and utilizes specific frameworks and architectures, primarily the Transformer model. Transformers, introduced by Vaswani et al. in 2017 [21], have become the backbone of many advanced NLP models due to their efficiency in handling sequential data and their ability to capture context through self-attention mechanisms. The key components of Transformers include the self-attention mechanism, which allows the model to weigh the importance of different words in a sentence relative to each other, capturing dependencies and context more effectively. Positional encoding is used since Transformers do not inherently understand the order of words, providing the model with information about the position of words in a sentence. Feed-forward neural networks are applied to each position separately and identically, processing the information after the self-attention mechanism. The Transformer architecture consists of an encoder to process input sequences and a decoder to generate output sequences, though some models use only the encoder or decoder.

The first step in developing a generative AI model involves collecting a large and diverse dataset of text relevant to the task at hand. This dataset needs to be preprocessed to remove noise, handle missing values, and standardize formats. Common preprocessing steps include tokenization (breaking down text into individual words or sub-words), lowercasing, removing punctuation, and handling special characters. Based on the task, the appropriate model architecture is selected. For generative tasks, the Transformer architecture is commonly used, with models like GPT (Generative Pre-trained Transformer) specifically designed for text generation tasks [25].

Training a generative AI model involves two main phases: pre-training and fine-tuning. During pre-training, the model is trained on a large corpus of text in an unsupervised manner, learning to predict the next word in a sentence, thereby understanding grammar, facts about the world, and some level of reasoning ability. The objective is to minimize the difference between the predicted word and the actual next word in the sequence. After pre-training, the model is fine-tuned on a smaller, more specific dataset tailored to the desired application. Fine-tuning helps the model adapt to the specific nuances and requirements of the target task, such as conversational

responses or creative writing. This phase typically uses supervised learning, where the model is trained with input-output pairs [26].

The trained model is then evaluated using metrics relevant to generative tasks. Common metrics include perplexity (a measure of how well the model predicts a sample), BLEU score (for machine translation), and human evaluations for subjective assessments of text quality, coherence, and relevance. Once the model performs satisfactorily on evaluation metrics, it can be deployed for inference. During inference, the model generates text by sampling from the probability distribution of the next word given the previous words. Various techniques, such as beam search, temperature sampling, and top-k sampling, can be used to control the diversity and quality of the generated text [26]. For example, developing a conversational AI model would start with collecting a large corpus of conversational data from sources like chat logs, dialogues, and transcriptions of spoken conversations. The text data is then cleaned and tokenized, handling contractions and removing irrelevant content. A pre-trained model like GPT-3, which is specifically designed for text generation, is chosen and fine-tuned on the conversational dataset to learn the nuances of human dialogue. The model's responses are evaluated using metrics like perplexity and human evaluations, and once satisfactory, the model is deployed in a chatbot framework, integrating it with user interfaces and back-end systems for real-time interaction.

Generative AI in NLP leverages sophisticated models like Transformers to create human-like text. The development process involves data collection and preprocessing, model architecture selection, training (pre-training and fine-tuning), evaluation, and deployment. By understanding and implementing these steps, one can build powerful generative models capable of performing a wide range of NLP tasks, from conversational agents to creative writing assistants.

1.4. Deep Learning, NLP & Biology

Deep learning has revolutionized various fields, including biology, where it has enabled significant advancements in understanding complex biological systems. Deep learning models originally developed for natural language processing (NLP) have been successfully adapted to tackle challenges in proteomics and genomics. These models leverage the power of neural networks to analyze and interpret large-scale biological

data, providing insights that were previously unattainable [27]. Deep learning models, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformers, have been pivotal in processing biological data. CNNs, known for their ability to capture spatial hierarchies, have been widely used in genomics for tasks like identifying regulatory motifs in DNA sequences. RNNs, designed to handle sequential data, are applied in analyzing time-series data such as gene expression profiles. However, transformers, with their attention mechanisms, have shown the most promise in capturing long-range dependencies in biological sequences, like their success in NLP tasks [28].

In proteomics, where the primary focus is on studying the structure and function of proteins, deep learning has been instrumental. One of the notable applications is protein structure prediction. Traditional methods like X-ray crystallography and nuclear magnetic resonance (NMR) spectroscopy are time-consuming and expensive. Deep learning models, such as AlphaFold [29] developed by DeepMind, have drastically reduced the time required to predict protein structures with remarkable accuracy. AlphaFold uses a deep neural network to predict the distances between pairs of amino acids, which are then used to construct the 3D structure of the protein. This model has achieved unprecedented success in the Critical Assessment of Protein Structure Prediction (CASP) competitions, outperforming all other methods.

Another significant application in proteomics is protein-protein interaction (PPI) prediction. PPIs are crucial for understanding cellular processes and disease mechanisms [30]. Deep learning models, like DeepPPI [31], utilize neural networks to predict interactions based on protein sequences. These models are trained on known PPI datasets and can be generalized to predict interactions in unseen data. The use of attention mechanisms in transformers allows these models to capture relevant features from long protein sequences, improving prediction accuracy.

Another notable model is DeepVariant [32], developed by Google, which employs deep learning for variant calling in DNA sequencing data. DeepVariant uses a CNN to process raw sequencing data and accurately identify genetic variants, such as single nucleotide polymorphisms (SNPs) and insertions/deletions (indels). The model has demonstrated superior accuracy compared to traditional methods, improving the reliability of genomic analyses and aiding in precision medicine.

Despite these advancements, several challenges remain in applying deep learning to biology. One major challenge is the limited availability of high-quality annotated data for training models. Biological data is often noisy and heterogeneous, requiring careful preprocessing and curation. Additionally, the interpretability of deep learning models is a significant concern. Understanding how these models make predictions and identifying the biological relevance of the learned features is crucial for gaining trust and acceptance in the scientific community [33].

Deep learning has had a profound impact on biology, particularly in the fields of proteomics and genomics. Models originally developed for NLP tasks, such as transformers, have been successfully adapted to analyze biological sequences, leading to breakthroughs in protein structure prediction, variant calling, and gene regulation analysis. Despite the challenges, the continued development and application of deep learning models hold great promise for advancing our understanding of complex biological systems and driving innovations in precision medicine and biotechnology. By leveraging the power of deep learning, researchers can uncover new insights, identify novel therapeutic targets, and ultimately improve human health.

1.5. Problem Statement and Solution

1.5.1. Problem statement

Despite significant advancements in deep learning and its application to biotechnology, the generation of accurate and effective antibody sequences remains a complex and resource-intensive task. Traditional methods of antibody development are costly and time-consuming, often requiring extensive experimental procedures. Recent deep-learning models have shown promise in leveraging the inherent information within protein sequences for antibody design, but there is still a need for models that can generate antigen-specific antibodies with higher accuracy and efficiency.

1.5.2. Solution

To address the challenges associated with generating accurate and efficient antigen-specific antibody sequences, this research proposes utilizing a pre-trained deep learning model that leverages state-of-the-art natural language processing (NLP) architectures. The pre-trained model, originally trained on extensive protein sequence

datasets, will be fine-tuned specifically on antibody data to enhance its capability in generating antigen-specific accurate antibody sequences. By using full-length antigen sequences as input, the fine-tuned model aims to significantly improve the accuracy of sequences. This approach seeks to provide a more reliable and resource-efficient method for antibody design, reducing the reliance on traditional, resource-intensive experimental methods and accelerating the discovery of potent antigen-specific antibody candidates.

1.5.3 Objectives

This research aims to develop a transformer-based deep-learning model to generate antigen-specific antibody sequences. The objectives include:

- Developing a transformer-based model architecture tailored for antigen-specific antibody sequence generation by fine-tuning a pLM.
- Evaluating the model's ability to generate high-affinity, antigen-specific antibody sequences.

By achieving these objectives, the research seeks to address the current limitations in antibody sequence generation and provide a more efficient approach to developing therapeutic antibodies.

Chapter 2: LITERATURE REVIEW

Earlier different types of deep learning models have been discussed along with their architecture. In this section, different deep learning models for antibody development, their architecture, and results that are reported in the literature will be discussed. Our focus will be on language models, especially the ones that generate protein or antibody sequences. These models are known as protein language models (pLM) [27], most of them are transformer-based models and pre-trained on large data sets of protein sequences. These models are discussed in the following section. The application of DL in this area leverages vast datasets of protein and antibody sequences, combined with structural and functional annotations, to train models that can predict and generate new sequences. This advancement holds promise for accelerating drug discovery and improving therapeutic efficacy.

2.1. Transformers in NLP Task

In the paper "Transformers: State-of-the-Art Natural Language Processing [34]" by Thomas Wolf and his team at Hugging Face, the focus is on how transformer architectures have revolutionized the field of natural language processing (NLP). Introduced in 2017, transformers have overtaken previous models like convolutional and recurrent neural networks by handling large-scale data more efficiently, training in parallel, and recognizing complex features in text sequences.

The main contribution of this paper is the introduction of the "***Transformers***" library. This open-source resource has been a game-changer by making cutting-edge NLP models accessible to a broader community. The library supports a wide range of NLP tasks through its extensive collection of pre-trained models, tokenizers, and datasets designed to be adaptable for both research environments and real-world applications.

One of the core challenges discussed in the paper is how to effectively implement these models across different platforms, ensuring they are scalable and deployable in various settings. The "Transformers" library addresses these issues by providing tools that simplify the training, scaling, and deployment processes, making it easier for users to leverage transformer models for their specific needs.

This paper is essential for understanding how the "Transformers" library is democratizing advanced NLP technologies, enabling users to achieve state-of-the-art results in various NLP tasks with relative ease. The insights from this study highlight the potential of transformer models to enhance and streamline NLP applications, setting a new standard in the field.

2.2.Innovations in Generative Modelling for Protein Design

The field of developing protein language models is excelling at such a pace that every day new models are being reported in scientific journals and model hubs. The very first protein models that used simple neural network architecture to generate protein sequences were reported in 2018 with the title, “Computational Protein Design with Deep Learning Neural Networks” by Jingxue Wang and colleagues [35]. They developed a multi-layer neural network model to predict the likelihood of 20 natural amino acids at each residue position within a protein, based on a comprehensive dataset of protein structures. By incorporating various structural properties as input features, the model achieved an accuracy of 38.3%.

This research demonstrates the significant role of deep learning in capturing complex patterns within large datasets, which traditional methods might miss. By applying the network's output as residue type constraints, they enhanced the accuracy of protein designs using Rosetta software for three specific proteins. Their results not only showed improvements over previous methods but also highlighted the neural network's ability to contribute effectively to the field of protein design.

These findings are vital as they provide a new perspective on utilizing machine learning to refine and accelerate the computational design of proteins. The implications of this study suggest that deep learning could play a crucial role in developing proteins with desired functions and structures, potentially leading to advancements in medical, biological, and chemical applications.

Meantime, Transformer architecture got the attention of researchers, and they started to build transformer-based architecture and trained them on protein sequence data. The paper titled "ProtGPT2 is a deep unsupervised language model for protein design" delves into the capabilities of ProtGPT2, an advanced language model specifically crafted for protein design. This model, utilizing the transformer

architecture, is trained on an extensive dataset of protein sequences and employs autoregressive training methods to generate new, de novo protein sequences that are evolutionary distant yet functionally viable [36].

ProtGPT2 distinguishes itself by its ability to produce protein sequences that are not only structurally akin to natural proteins but also explore uncharted regions of the protein space, known as 'dark' areas. This is critical for expanding the diversity of protein structures available for scientific study and application. The generated proteins predominantly exhibit globular structures with appropriate disorder and secondary structure content that aligns closely with those found in nature. A key aspect of ProtGPT2's functionality is its potential in high-throughput settings where it can generate sequences rapidly, making it an invaluable tool for both research and practical applications in fields ranging from biomedicine to environmental science. This model's ability to generate proteins that can potentially fold into stable, ordered structures similar to those observed in nature underscores its utility in advancing the protein engineering field.

There are some other models too that use transformer/attention-based architecture for protein sequence modelling. For example, "Transformer-based Protein Generation with Regularized Latent Space Optimization" (ReLSO) [37] introduces a novel approach to protein design, combining transformer-based autoencoders with optimization techniques in a structured latent space. In this method, Castro et al. address the complexity of protein design by effectively navigating a vast space of potential amino acid sequences, allowing for the optimization of specific properties like stability and binding affinity through gradient ascent. ReLSO enhances the efficiency of protein sequence exploration by avoiding local maxima, a common challenge in traditional methods, and employs regularization techniques to maintain a favorable fitness landscape. This streamlined approach significantly reduces the experimental demands of protein engineering, merging advanced AI strategies with biological research to foster developments in biotechnology and therapeutic design.

Shin et al. (2021) [38] and Shuai et al. (2021) [39] have both leveraged generative models to enhance protein and antibody design. Shin et al. developed alignment-free autoregressive models that synthesize protein sequences without evolutionary constraints, this study is important to generate diverse protein sequences

and bypassing the other computational methods that rely on multiple sequence alignment. Concurrently, Shuai et al. focused on generating full-length antibody sequences with their IgLM-designed libraries, showing favourable outcomes over naive baselines. However, the models' performance near variable regions like CDRs still requires improvement.

2.3.Deep Learning for Specificity and Optimization in Antibody Design

Mason et al. (2021) introduced a deep-learning model that predicts antigen specificity and optimizes antibody variants, marking a significant advancement in therapeutic antibody development. They generated many antibodies for HER2 by using the mutagenesis technique and then classified them as a binder and non-binder. They trained CNN on binding data and proposed their model as the optimizer for HER2. Their model, specifically effective for HER2, showcases the potential of machine learning in narrowing down viable antibody candidates based on complex biological parameters [40].

2.4.Antibody Engineering Model

After promising results of different protein language models, the researchers started to train the models on antibody data to get more reliable, specific, and efficient potential candidates. For this purpose, different deep learning architectures were proposed that were pre-trained on different datasets of antibodies. These models have shown significant results and served as potential AI tools in antibody development and antibody sequence modeling. Below are some deep learning models that coined the term antibody models that have merged in the consequence of AI’s development [41], [42].

The first study to discuss here was published in a scientific report by Saka *et al.* (2021). They used LSTM-based architecture for antibody design. This approach leverages a phage display library to enhance the affinity maturation of antibodies against specific targets. By integrating LSTM models with next-generation sequencing (NGS) data, the research team efficiently identified high-affinity antibody sequences, significantly streamlining the traditional, labor-intensive process of affinity maturation. This method reduced the time and cost associated with antibody development but also

improved the potential to find high-affinity antibodies, which is critical for therapeutic and diagnostic applications [43].

Another famous work was done by big brains of John Hopkin's University and the University of California, Berkley, W. Shuai, *et al.* proposed the model Immunoglobulin Language Model (IgLM), [39] a deep generative language model optimized for designing synthetic antibody libraries. This model leverages autoregressive sequence generation techniques, like text infilling in natural language processing, to enhance the development of monoclonal antibodies. IgLM is distinct in its ability to redesign variable-length spans of antibody sequences, which significantly accelerates the discovery of therapeutic antibody candidates. The model, trained on over 558 million antibody heavy and light chain variable sequences, uses this vast dataset to generate synthetic libraries that display desirable biophysical properties consistent with those of natural antibodies, but with lower immunogenicity and increased "humanness." These properties are crucial for reducing the likelihood of rejection in therapeutic applications. The study showcases IgLM's capability to generate full-length antibody sequences and diversify loops within these sequences, maintaining high sequence fidelity to known antibody structures while enabling rapid exploration and innovation in antibody design.

Another study recently published in "Genomics and Proteomics Bioinformatics" by Xiaopeng XU and colleagues represented the model "AB-Gen" in the study "AB-Gen: Antibody Library Design with Generative Pre-trained Transformer and Deep Reinforcement Learning". They introduced an innovative method using reinforcement learning and generative transformers to optimize antibody design, used a trained transformer model as the policy network in a reinforcement learning framework to explore and generate antibody sequences, particularly focusing on the heavy chain complementarity-determining region 3 (CDRH3). The method improved the design process by generating sequences that meet multiple property constraints, such as specificity to human epidermal growth factor receptor-2 (HER2), solubility, and reduced immunogenicity.

AB-Gen's effectiveness is demonstrated by its ability to produce novel CDRH3 sequences that not only exhibit desirable biophysical properties but also surpass traditional design methods in terms of success rate. The use of deep learning

significantly reduces the experimental burden by filtering out suboptimal candidates early in the design process, thus enhancing the efficiency and outcome of preclinical antibody discovery and development.

The reviewed studies demonstrate significant advancements in using artificial intelligence for protein and antibody design, showing that these technologies can speed up the creation of new biomolecules with potential therapeutic uses. However, these studies also face limitations, such as high computational costs and dependence on extensive and diverse data sets, which can affect the models' effectiveness and accessibility. Additionally, integrating these computational tools into existing scientific workflows remains challenging, often requiring substantial experimental validation to confirm the computational predictions. Moving forward, improving the efficiency, data handling, and integration of these models will be crucial to fully realize their potential in practical applications.

2.5.Antigen-specific models for Antibody Engineering

Until now the models that are discussed in the above section and a few others that were architected for antibody sequence generation are proficient antibody-like sequences that preserve the properties of antibodies, their efficacy, and specificity in general. But an important aspect to note is that antibodies in nature are produced in response to antigens. Meaning when antigens enter the body the immature B cells produce antibodies against these pathogens. Similarly, there should be an intelligent system that acts like the natural system to produce antibodies against the antigens. After an in-depth exploration of the literature, only a few such studies were identified that incorporate the antigen/epitope information to model antibody sequences. Two out of them are discussed below.

The first study "Epitope-specific antibody design using diffusion models on the latent space of ESM embeddings" by Tomer Cohen and Dina Schneidman-Duhovny introduces EAGLE (Epitope-specific Antibody Generation using Language model Embeddings), a novel approach for designing antibodies [44]. Unlike conventional methods that rely on a predefined antibody structure, EAGLE employs a diffusion process, akin to generative models in image processing, conditioned on the antigen's structure and specific epitopes. This allows for the design of antibodies directly in a

continuous latent space provided by protein language model embeddings, enabling the creation of diverse and unique antibodies with variable loop lengths.

EAGLE demonstrates substantial advancements by generating antibody sequences that exhibit up to 55% identity to known binders, especially in the most variable heavy chain loop, without the need for a starting structure. This capability is crucial for developing therapies targeting specific antigen aspects, such as conserved viral regions, potentially enhancing therapeutic efficacy and reducing resistance development. The model's performance is evaluated through metrics like Variable Length Amino Acid Recovery (VAAR) and docking scores, with EAGLE showing superior performance compared to other methods like HERN, which relies on fixed-length CDRs. This research signifies a significant step forward in computational biology, applying advanced AI techniques to the complex challenge of epitope-specific antibody design. By leveraging generative models conditioned on detailed biological information, the study not only enhances the understanding and capability of antibody design but also opens new avenues for more effective and targeted therapeutic interventions.

And the other study was done in 2023 "Pre-training Antibody Language Models for Antigen-Specific Computational Antibody Design" by Kaiyuan Gao et al. introduces AbBERT, a pre-trained antibody language model that revolutionizes computational antibody design [45]. By leveraging extensive sequence data, this model overcomes the limitations of scarce structural data and focuses on the highly variable complementarity-determining region (CDR), essential for binding specificity and affinity. AbBERT uses a one-shot generation technique to produce all amino acids of the CDR simultaneously, reducing the errors associated with traditional sequential methods. This innovative integration enhances both the sequence prediction and structural fidelity of the designed antibodies, ensuring their functionality and relevance in practical applications. The effectiveness of AbBERT is particularly notable in optimizing CDR-H3 regions, crucial for antigen binding, with superior performance against challenging targets like SARS-CoV-2. The model's ability to produce structurally accurate antibodies with high specificity is demonstrated through its excellence in docking scores and Variable Length Amino Acid Recovery (VAAR), consistently outperforming existing methods. This paper highlights a significant

advancement in antibody design, showing how AI-driven models like AbBERT can transform therapeutic developments by optimizing the design process for specific antigens, thus offering a promising avenue for creating targeted and efficient therapies.

These studies incorporated the antigen/epitope information in their model as input and generated the antibody CDR sequences and structures but were limited to short lengths of input like only the epitope region and only giving the CDRs as output.

2.6. Significance of the Study

The development of an antigen-aware generative model holds significant potential for the field of antibody engineering. Such a model could revolutionize the antibody development cycle by providing a means to rapidly generate candidate antibody sequences with high specificity and affinity for target antigens. This capability would streamline the initial stages of antibody development, reducing the time and cost associated with experimental screening and optimization.

By addressing the current limitations in antibody sequence generation, this research aims to make a significant contribution to the field of computational antibody engineering, paving the way for more efficient and targeted therapeutic developments. The successful implementation of an antigen-aware generative model would represent a major advancement in the ability to design and develop antibodies, with wide-ranging implications for healthcare and biotechnology.

CHAPTER 3: METHODOLOGY

3.1. Methodology Overview

In this section, the methodology to achieve the objective of this study is discussed briefly. The first step is data acquisition, followed by data curation. In the data curation step the data was curated to feed to the model. The next step was the selection of an appropriate model, finetuning the selected model on curated data and then finally model's output was evaluated using the Amino Acid Recovery Rate between the generated antibody and antigen was performed. All these steps are discussed comprehensively below.

3.2. Data Acquisition

A secondary dataset was used for this study collected from the SAbDab database [46] following the objective of this study only data from proteins as the antigen type were retrieved the data was in a protein data bank (pdb) file containing the antibody heavy and light chain structures and corresponding antigen structure. Along with the pdb files summary files were also retrieved that were used for the curation in the later step. The antibodies in this dataset were renumbered by the IMGT numbering scheme [12], [46].

3.3. Data Curation

This study is focused on sequences of antibodies and corresponding antigens, pdb files were subjected to amino acid sequence extraction and water removal using BioPython modules. Once the sequences were extracted, each pdb file had one or more antibody heavy and light chain sequences and corresponding antigen sequences. These amino acid sequences are labeled with the chain IDs e.g., Chain A, Chain B, Chain C. The annotation of these chains is present in the summary file downloaded earlier. This summary file was used to curate the data that the CSV files of antibody sequences of heavy and light chains along with corresponding antigen sequences in a paired manner.

To achieve this data curation different Python libraries were used mentioned in the table given below.

Table 3.1 Python libraries used for data curation.

Library	Purpose
BioPython	A collection of tools for biological computation, providing libraries and scripts for working with bioinformatics data.
os	Provides functionalities for interacting with the operating system, such as file and directory manipulation.
CSV	Enables handling of CSV (Comma-Separated Values) files, including reading and writing operations.
Bio.SeqIO	Supports input and output operations on biological sequence data in various file formats.
PDBParser	Facilitates parsing and extracting structural information from Protein Data Bank (PDB) files.
Bio.Seq	Represents and manipulates nucleotide or protein sequences in BioPython.
Bio.SeqRecord	Manages sequences along with associated metadata, annotations, and features within BioPython.

3.4. Model Selection

After conducting a comprehensive literature review and gaining an in-depth understanding of various models, it was concluded that a sequence-to-sequence (Seq2Seq) model would be well-suited to achieve the objectives of this study. Seq2Seq models are extensively utilized in natural language processing tasks, including machine translation. These models typically employ an encoder-decoder architecture, where the encoder processes the input sequence and converts it into a context vector. This context vector is then provided to the decoder, which generates the output sequence iteratively. AlphaFold is a transformer-based Seq2Seq model, which takes protein sequence as input and generates the 3-D spatial arrangement of its amino acids. The task was approached as a machine translation problem, where antigen sequences are provided as input to the model, and antibody sequences are generated as output. The Text-to-Text Transfer Transformer (T5), a versatile and robust sequence-to-sequence model

developed by Google Research, was employed for this purpose. Its text-to-text framework made it a strong candidate to apply in protein space. This was attempted by Elnaggar *et al*, who developed several different models one of them is **prot_t5_xl_uniref50(protT5)**, pre-trained on 45 million protein sequences of the Uniref50 dataset [47].

3.4.1. protT5

ProtT5 uses an encoder-decoder structure typical of the T5 architecture. The encoder processes input sequences and generates context-rich embeddings, which the decoder then uses to produce output sequences. This architecture is particularly well-suited for tasks that require generating sequences, such as predicting protein structures or functional annotations. The ProtT5 model comes in two sizes: ProtT5-XL and ProtT5-XXL, with 3 billion and 11 billion parameters, respectively. These models were trained using a combination of UniRef50 and BFD (Big Fantastic Database) datasets. The initial training on the larger, more diverse BFD dataset was followed by fine-tuning on the more curated UniRef50 dataset. This two-step training process enhances the model's ability to generalize and adapt to specific protein-related tasks.

3.4.2. Architecture of protT5

ProtT5 utilizes the "T5ForConditionalGeneration" architecture, optimized for tasks that involve generating sequences from input data. This model is characterized by a high-dimensional setting, with a model dimension (d_model) of 1024 and a feed-forward layer dimension (d_ff) of 16384, allowing it to capture complex patterns within protein sequences. The model incorporates a robust attention mechanism with key, query, and value vectors (d_kv) sized at 128, facilitating focused processing of sequence features. To combat overfitting, a dropout rate of 0.1 is employed, alongside ReLU activations within the feed-forward projections to enhance non-linear learning.

ProtT5 is structured as a seq2seq model, featuring both encoder and decoder segments that process and generate sequences, respectively. It supports up to 512 positions in sequence length and utilizes 32 attention heads to manage multiple sequence relationships. The model’s vocabulary size stands at 128, allowing it to

efficiently handle diverse protein-related data. This configuration highlights ProtT5's advanced capabilities in analyzing and predicting complex biological sequences.

3.5. Model Fine-tuning

For the finetuning of the original protT5, the model was loaded which is available as open source on the Hugging Face platform. From this original model, two models were fine-tuned, the Light-chain model and the Heavy-chain model, separately. The two finetune models were trained using the transfer learning technique. The following steps were followed for transfer learning.

- **Loading a Pre-trained Model:** The pre-trained model, ProtT5, was utilized to leverage previously learned features from extensive training on a large dataset with 3 billion parameters.
- **Customizing the Configuration:** The configuration of the model was customized by modifying its setup to reduce its complexity, including decreasing the number of layers. For the heavy chain model, 4 encoder and decoder blocks with 4 attention heads were used along with 3384 feed-forward layers. For light chain model, 6 encoder and decoder blocks with 6 attention heads along with 16684 feed-forward layers. For both models positional encoding of 512 was used.
- **Preparing Data with a DataLoader:** The data was prepared using a DataLoader to ensure that it was in a format suitable for processing by the customized model, along with train and test splits with an adequate batch size of 370.
- **Training the Customized Model:** The customized model was trained on a SabDab and AbDb paired antigen-antibody sequences. to update its weights and biases.
- **Saving the Model:** After training, the model was saved to preserve its state, including all learned weights.

3.5.1. Cross Entropy Loss

Cross-entropy loss is a performance metric that quantifies the difference between two probability distributions: the predicted probability distribution and the actual distribution of the labels. In machine learning, especially in classification tasks,

it is used to measure the error between the model's predicted outputs and the actual target values.

Adapted Cross-Entropy Loss Formula

For such applications, the sequence-based cross-entropy loss is computed as follows:

$$\mathrm{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T_i}\sum_{c=1}^{C} y_{i,t,c}\, log\left(p_{i,t,c}\right)$$

Where:

- N represents the number of samples, typically corresponding to the batch size during training.
- T_i denotes the length of the sequence for the *i-th* sample.
- C is the total number of classes, equivalent to the size of the vocabulary in language models.
- $Y_{i,t,c}$ is a binary indicator (0 or 1), specifying whether class label ccc is the correct classification for the *t-th* token in the *i-th* sequence.
- $p_{i,t,c}$ represents the predicted probability that the *t-th* token in the *i-th* sequence belongs to class ccc.

This formula ensures that the loss is calculated across all tokens in the output sequence, comparing the predicted probabilities at each token position against the actual token labels in a multi-class setup. By computing the loss in this manner, the model is trained to enhance its accuracy across the entire sequence, which is essential for achieving high performance in sequential prediction tasks.

Importance in Training

The sequence-based cross-entropy loss is integral to the training process as it provides a granular measure of the model's prediction accuracy at each token position within the sequences. This loss function penalizes incorrect classifications harshly, especially when the model assigns low probabilities to the correct classes, thereby encouraging significant adjustments in the model's parameters. The logarithmic nature of the loss function amplifies the penalties for errors, ensuring that the model not only predicts accurately but also with high confidence across the length of the sequence.

This methodological approach to loss calculation is crucial for developing robust models capable of handling complex sequence generation tasks, thereby ensuring that the models perform optimally in real-world applications where accuracy and reliability are paramount.

3.6.Training

For training purposes, the Pytorch framework was used along with other important libraries like Transformer, sklearn, numpy, and pandas. A detailed overview of libraries that were imported for the training is given in Table 3.2. All these packages were utilized efficiently to achieve the model training.

Table 3.2 List of Libraries/modules used in full finetuning of ProtT5.

Library/Module	Usage
CUDA 12.1	To train the model on a Nvidia GPU device, Pytorch utilizes CUDA for faster computations during model Training.
huggingface-hub 0.19.4	This package was used to download the ProtT5 original model from Hugging Face.
Jupyter 1.0.0	Jupyter Notebooks provides an interactive workspace for code execution along with the visualization of plots.
NumPy 1.26	Used for numerical computations as they are an integral part of model training.
Pandas 2.1.3	Enables powerful data manipulation and analysis, providing DataFrames and tools for working with structured data.
PyTorch 2.1.1	Facilitates the creation, training, and optimization of neural networks with dynamic computational graphs, widely used for deep learning tasks.
Scikit-learn	Offers tools for data preprocessing, model evaluation, and splitting datasets into training and testing sets, essential in machine learning workflows.
Seaborn	Used for creating attractive and informative statistical graphics, often in conjunction with Pandas and NumPy for data visualization.
Streamlit	Enables the rapid development of web applications for machine learning models, allowing for easy deployment and sharing.
TensorFlow 2.15	Serves as a deep learning framework for training complex neural networks, often used in conjunction with Keras for building models.

Tokenizers 0.15	Provides efficient tokenization of text data for NLP tasks, used with transformers for handling large datasets in natural language processing.
TQDM 4.66	Adds progress bars to loops in Python, which is useful for tracking the progress of long-running operations such as model training.
Transformers 4.35	Provides pre-trained models and tools for fine-tuning them on specific tasks, particularly in natural language processing and other AI domains.

All the above libraries were utilized, and the training of the model was ensured in this study. The process of finetuning the model and the usability of packages is discussed as follows.

3.6.1. Setup and Configuration

The study utilized the PyTorch framework and the Hugging Face Transformers library to configure and train a T5 model specifically tailored for translating antigen chain sequences into heavy chain sequences. Initial setup verified GPU availability to utilize CUDA 12.1, enhancing computation efficiency critical for deep learning models. A custom T5 model configuration was defined (T5Config), with parameters set to optimize the model for the specific requirements of the task. These parameters included adjustments to the dropout rate, activation functions, and the model's architecture to accommodate the sequence complexity and length encountered in the dataset.

3.6.2. Data Preparation

Data was sourced from a CSV file containing antigen and heavy chain sequences. This dataset was pre-processed to remove any entries with missing data (*dropna*) and to ensure all text data was in string format. Stratified splitting was employed to divide the dataset into training (80%), validation (10%), and test (10%) sets, maintaining a consistent distribution of data across each set for unbiased model training and evaluation.

3.6.3. Tokenization and Data Loading

The T5Tokenizer was utilized to tokenize the text data. This tokenizer converted sequences into token IDs, compatible with the T5 model's input requirements. Batch

encoding (***batch_encode_plus=True***) was applied to transform both source and target texts into padded token tensors, facilitating uniform input sizes for batch processing. Custom Dataset and *DataLoader* classes were implemented to manage this tokenized data efficiently, ensuring that data was shuffled for the training set to prevent model overfitting and fed in batches to the model during training.

3.6.4. Model Initialization and Multi-GPU Setup

The T5 model was instantiated from the pre-trained state with the specified custom configuration and then transferred to the appropriate computational device (GPU). For environments with multiple GPUs, ***DataParallel*** was employed to distribute the model's workload across available GPUs, significantly speeding up the training process.

3.6.5. Training Loop

The model training was conducted over 30 and 15 epochs for heavy and light chain models respectively. Each epoch consisted of a forward pass where the model predicted the heavy chain sequence from the antigen chain sequence, followed by a backward pass where the model's parameters were updated based on the computed loss. Loss calculation was performed using the model's outputs against the ground truth sequences, with the ***AdamW*** optimizer and a learning rate of 0.001 schedulers managing the optimization process.

3.6.6. Evaluation and Checkpointing

After each training epoch, the model's performance was evaluated on both validation and test datasets. This evaluation assessed the model's translation accuracy and its ability to generalize. Metrics such as average loss and accuracy were computed and stored. Model checkpoints were saved at the end of each epoch, allowing the training process to be resumed from specific points and facilitating the selection of the best-performing model based on validation data.

3.6.7. Final Output

Upon completing the training epochs, the final model state was saved to a designated directory. This model encapsulated all learned parameters and configurations, optimized through rigorous training and ready for deployment or further fine-tuning. Validation and testing were critical phases conducted at the end of each training epoch, where the model's performance was rigorously evaluated on unseen data. These evaluations were performed within a no-gradient context to enhance computational efficiency, calculating both loss and accuracy to assess the model's generalizability and robustness.

Checkpointing was systematically implemented, saving the model's state along with the optimizer and scheduler states at the end of each epoch, 35 checkpoints were for heavy chain model and 15 checkpoints were for light chain model. This strategy not only safeguarded the training progress by providing recovery points but also ensured the retention of the best-performing model configuration throughout the training sequence. Upon the completion of the training regimen, the fully fine-tuned models for both the light and heavy chain configurations were saved to their respective paths.

This unified training approach provided a structured and replicable framework, critical for advancing research in protein chain analysis and ensuring that both models were optimized under consistent conditions for subsequent comparative and standalone evaluations.

3.7. Evaluation

The next in model development after training is evaluation, where the model is assessed using different evaluation metrics. Depending on the task and the model architecture these evaluation metrics vary from case to case. For this study, in which the output of the model is the generated sequence of antibodies three different types of evaluation metrics are used by different studies. These evaluation metrics validate the generated sequence by comparing them with the original antibody sequences against the test data set.

3.7.1. Amino Acid Recovery Rate (AAR)

AAR measures the proportion of amino acids in a predicted sequence that exactly matches those in the original sequence, at the correct positions. The formula for AAR, based on your description and code, is:

$$\text{AAR} = \frac{\text{Matching Amino Acids}}{\min(|\text{original\_sequence}|)} \times 100]$$

The ***AlignmentScore*** calculates the similarity score between two sequences by determining the total number of matching characters in the sequences when they are aligned optimally. ***AlignmentScore*** was calculated by using ***SequenceMatcher*** from Python's ***difflib*** module.

The alignment score S between two sequences seq1 and Seq2 can be mathematically represented as:

$$\text{S}(\text{seq1},\ \text{seq2}) \ = \sum_{\text{blocks}} \text{n}$$

where each block in the list of matching blocks provided by "SequenceMatcher" contributes "*n"* matched characters to the total score

AAR is then calculated by dividing this score by the maximum length of either the original or recovered sequence, which normalizes the score by the longest sequence to account for differences in length. In the context of antibody sequence modelling, AAR is important because it evaluates how well the predicted sequence recovers the original sequence, larger the value of AAR, better the performance of the model is.

3.7.2. Variational Amino Acid Recovery Rate (VAAR)

VAAR extends the concept of AAR to handle sequences of variable lengths, which is common in regions like the Complementarity Determining Regions (CDRs) of antibodies. The formula for VAAR is:

$$VAAR(x, y) = \frac{\text{AlignmentScore}(x, y)}{\max(|x|, |y|)} \times 100$$

Similar to AAR, VAAR uses the ***AlignmentScore*** but adapts it for sequences where the length may vary significantly, such as between different CDRs. By normalizing against the maximum length of the two sequences being compared, VAAR provides a metric that reflects how well the variable-length sequences are recovered or replicated, despite these length differences. The ability to accurately replicate or predict variable-length sequences is crucial for designing antibodies with high specificity and affinity. VAAR is critical for evaluating how well a computational model can handle the inherent variability in antibody sequences, especially in the CDRs, which directly interact with antigens.

3.7.3. Sequence Identity (SeqID)

SeqID quantifies the percentage of identical amino acids between two aligned sequences, normalized by the length of the shorter sequence. The formula for SeqID is:

$$SeqID(x, y) = \frac{\text{AlignmentScore}(x, y)}{\min(|x|, |y|)} \times 100$$

This metric calculates the ***AlignmentScore*** divided by the length of the shorter sequence, which emphasizes how well the shorter sequence is recovered or represented in the alignment. Sequence Identity is an essential metric in bioinformatics for assessing the similarity between two sequences, indicating how much of the smaller sequence is preserved in the larger one. This is particularly important in validating how well predicted CDRs align with known CDRs.

3.7.4. Test Set

Model performance is typically evaluated using benchmark datasets that are commonly employed in related studies to facilitate comparative analysis of different methods. However, in the current study, no such benchmark dataset was available. Hence, the common practice of using unseen data for evaluation was utilized. This unseen data can be sourced from the same dataset as the training data or from other suitable sources deemed appropriate for evaluation purposes. In this study, the test data was obtained from the same source, SabDAb. All complexes that were added to the database after the retrieval of the initial training data were identified and downloaded

for use as the test set. There were more than 300 complexes added to the database, out of which 170 complexes were selected as the test set for this study.

CHAPTER 4: RESULTS

In this section, the results of the methodology followed for this study will be discussed comprehensively.

4.1.Data Acquisition

From the SAbDab database, a total of 5091 structures were downloaded along with their summary files that record data for each heavy-light chain pairing in each Protein Data Bank (PDB) structure. These files include the PDB code, chain identifiers for heavy and light chains (noted as 'H' and 'L' respectively, or 'NA' if unpaired), and model numbers, typically '0' for X-ray structures. Additional details cover the antigen chains and their types (such as protein or peptide), with specifics provided for bound or unbound states. The summary also provides a short header describing the molecule, deposition date, compound description, and biological sources including organisms and species of the antibody and antigen chains. Authorship, publication details, structural resolution, and determination methods are also noted. Structural features like engineering status and subclass information for chains are included alongside biophysical data such as antibody affinity, Gibbs free energy changes, affinity determination methods, and experimental conditions. This comprehensive data set provides essential insights into the antibody structures housed within the database.

4.2.Data Curation

The annotations provided in the summary files were utilized to curate the data, ensuring it was properly formatted for input into the model. Given that the base model is a T5-like architecture, which requires paired data, various Python libraries were utilized to restructure the data into a paired format, as outlined in Table 1.1. This step was crucial because the study employed a transfer learning approach, where a model pre-trained through unsupervised learning was subsequently fine-tuned using a supervised learning approach. Table 4.1 provides a representation of the data used.

Table 4.1 Representation of the paired dataset after curating to be used for training.

PDB_ID	Antigen_Sequence	Light_Chain_Sequence
5IKC_1	[A A K K V R F Y R N G D R Y G D R Y F K G, ...]	[D I V M T Q S Q K L M S T S V G D R V S I T C K A S Q I V D ...]
2I9L_1	[I V Y A V S S D R F R S F D A L L A D L T R S L ...]	[D I V M T Q S Q K L M S T S V G D R V S I T C K A S Q I V D ...]
5CZV_1	[R S L S D N I N L P Q G V R Y I Y T I D G S R ...]	[D I V M T Q S Q K L M S T S V G D R V S I T C K A S Q I V D ...]
6CBV_1	[K I G S M D E L E E G E S Y V C S S D N F F K K V E Y T K, ...]	[D I V M T Q S Q K L M S T S V G D R V S I T C K A S Q I V D ...]

4.3. Training

Model training was performed on the Nvidia P400 GPU, the light_chain model took approximately 6 days (6814m) to train on the data set. While the heavy chain model took almost 4 days to train on the data. During training the test and validation losses were calculated and are shown below.

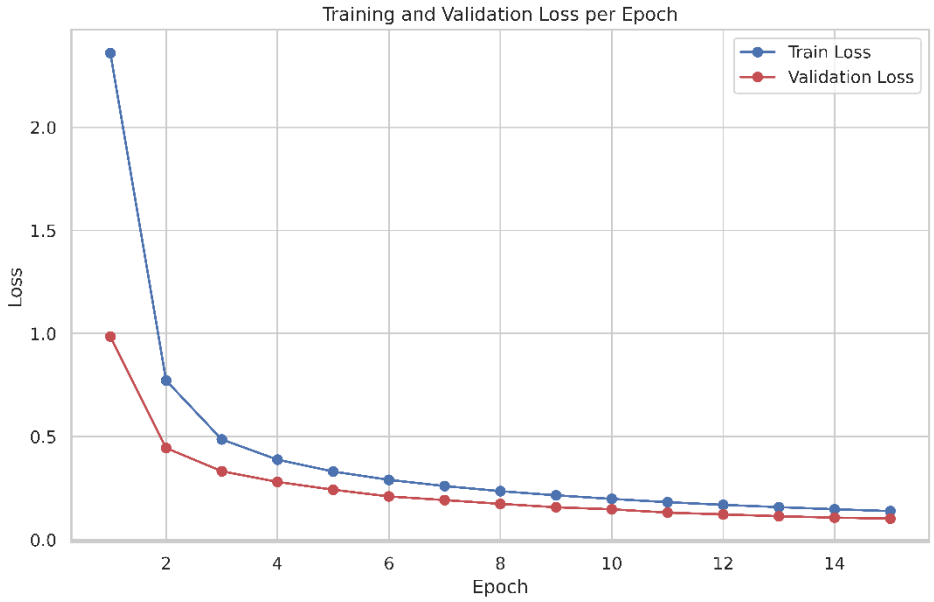

Figure 4.1 Training and Validation Losses for the light chain model calculated during training.

The above graphs showed a gradual decrease in training and validation loss across the epochs, which indicates that the model was learning efficiently during the training. The cross-entropy loss function was adaptively used for calculating the loss as described earlier in section 3.5. At the final epoch, the training loss was 0.138 and the validation loss was 0.102, at this point, the model was saved. For the heavy chain model, the same loss function was used as well, the training loss at the final checkpoint was 0.20 and the validation loss was 0.11. Figure 4.2 given below represents the training and validation loss during the training.

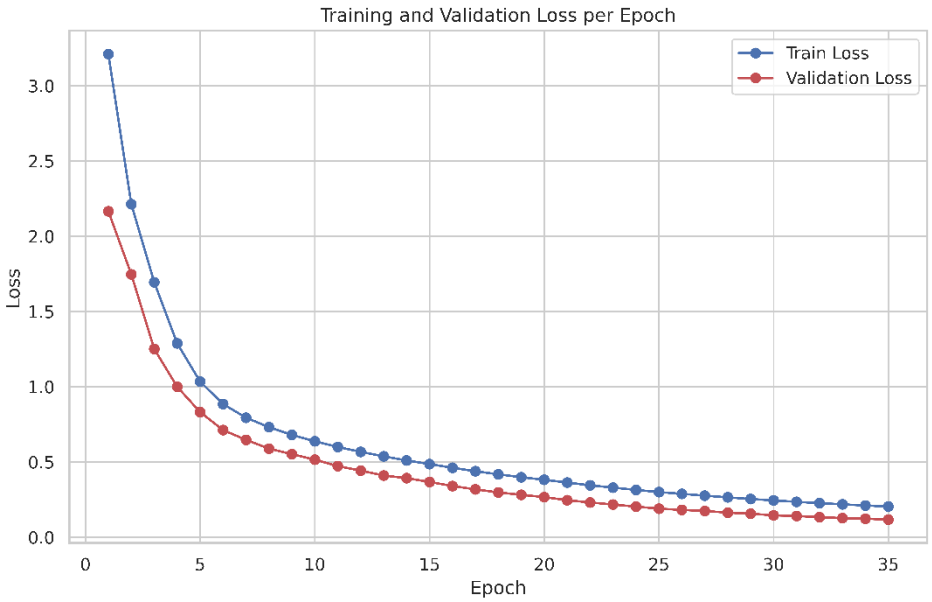

Figure 4.2 Training and validation loss of Heavy chain model calculated during the training.

Like the light chain model, the training and validation loss for the heavy chain model also showed a gradual decrease during the training. These graphs depict that the models are smoothly and efficiently trained on the data.

4.4. Evaluation

After the training, the models were saved and evaluated on different evaluation metrics. These evaluation metrics are AAR, VAAR, and SeqID mentioned in section 3.6 on the test set. For this purpose, the antigen sequence of the test set was used to generate antibody light chain and heavy chain sequences by loading the two respective models at the evaluation stage, and the inference was drawn. A dedicated function,

translate, is designed to handle the generation of both chains. This function first encodes input antigen sequences into a tensor format suitable for model input. Using the model's ***generate*** method, it then produces multiple sequence predictions, introducing variability and diversity through parameters such as *temperature*, *top-k*, and *top-p* sampling. These parameters are important because they help to explore the variation in the generated sequences. The value for these parameters can be adjusted to generate diverse sequences thus increasing the chances of potential antibody sequences which can be narrowed down after further downstream analysis like humanness score and solubility. By using the above-mentioned mechanism, 10 sequences were generated against each antigen sequence of the test set from both the models, and these generated/predicted heavy and light chain sequences were evaluated, the results of predicted sequences are given in the following sections.

4.4.1. AAR

A total of 1700 antibody light and heavy chain sequences were generated against 170 antigen sequences (10 sequences against each antigen), and AAR was calculated for both chains. This study reports very promising results with a mean AAR of ~ 60% and a maximum reported AAR is ~ 97% for Light chain sequences. For the heavy chain sequences, ~ 40% mean AAR is reported, and a maximum AAR of ~76%. The distribution of AAR for the test set is shown below in the histograms for both the light and heavy chain sequences.

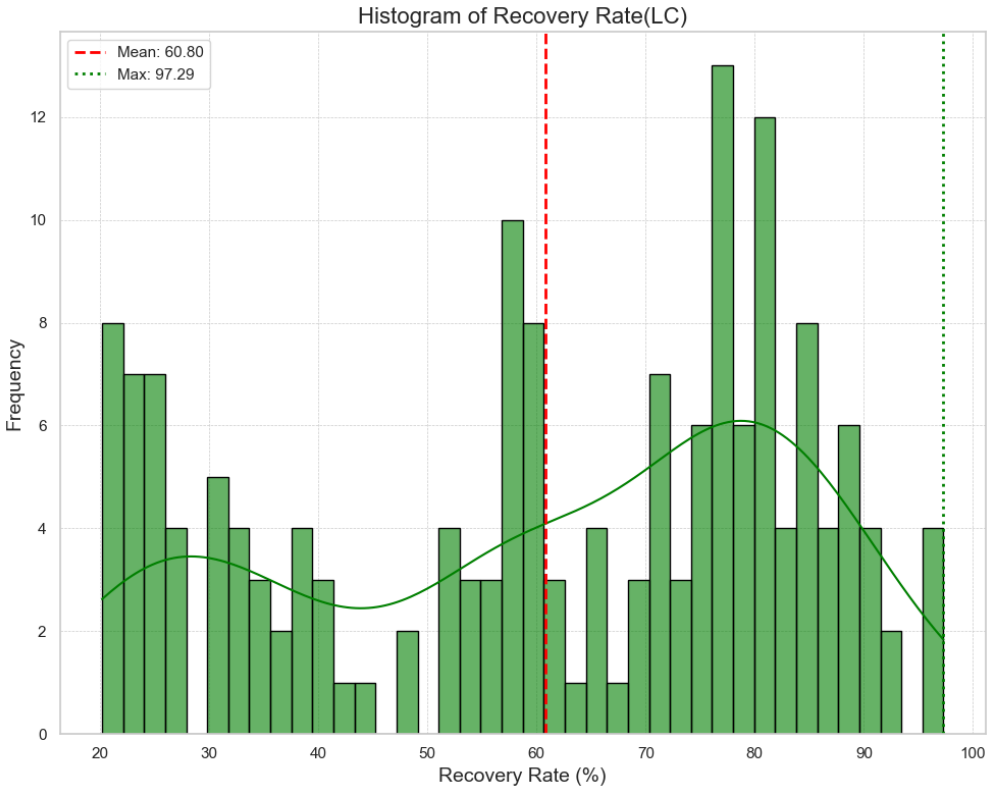

Figure 4.3 Histogram of Recovery Rate (%) for Light Chains (LC)

This histogram displays the distribution of recovery rates for light chain sequences. The red dashed line represents the mean recovery rate at 60.80%, while the maximum recovery rate is shown by a green dotted line at 97.29%. The distribution is bimodal, with significant peaks indicating high efficiency in recovery rates at both moderate and near-optimal levels, underscoring the enhanced recovery performance of light chain sequences for the test data set.

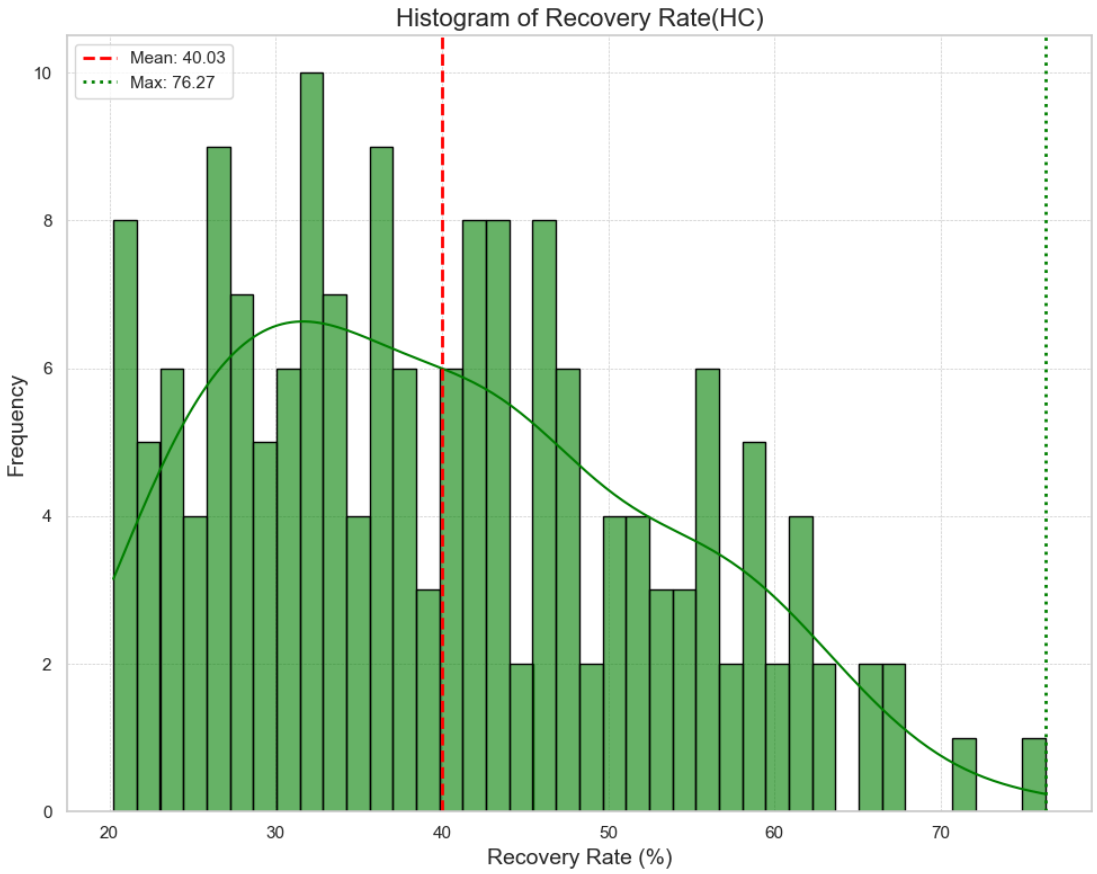

Figure 4.4 Histogram of Recovery Rate (%) for Heavy Chains (HC)

This histogram illustrates the distribution of recovery rates across heavy chain sequences. The mean recovery rate is denoted by a red dashed line at 40.03%, and the maximum recovery rate is marked by a green dotted line at 76.27%. The distribution primarily shows a single peak, reflecting a concentration of recovery rates around the mean, indicating moderate efficiency in the recovery of heavy chain sequences.

4.4.2. VAAR

To evaluate the variability between the original and predicted antibody sequences, VAAR was calculated. This study showed ~ 53% mean and ~ 96% VAAR for predicted light chain sequences across the test set. Similarly, for heavy chain sequences average VAAR of ~ 44% and maximum VAAR of ~ 76% was calculated. The distribution of VAARs for the chains is shown in the below histograms.

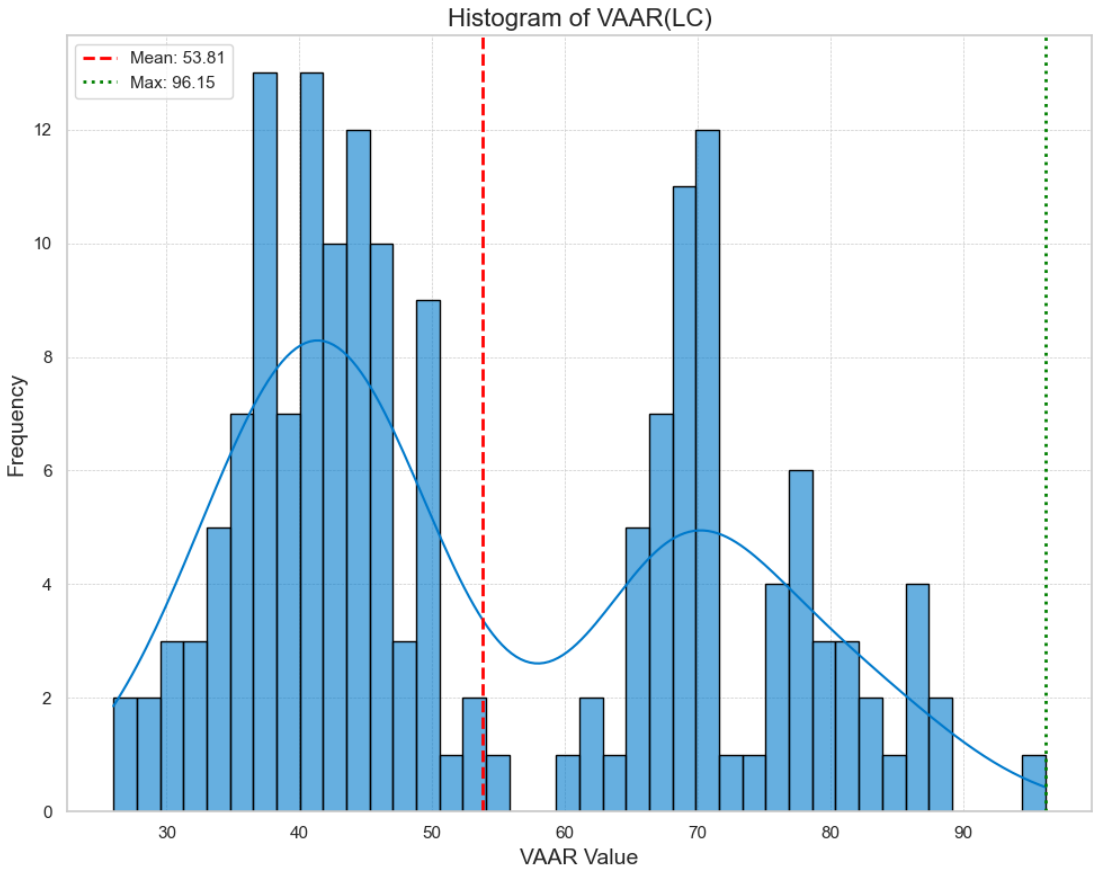

Figure 4.5 Histogram of VAAR (%) for Light Chain (LC)

This histogram displays the distribution of VAAR values for light chain sequences. The distribution is shown with a mean VAAR of 53.81%, indicated by the red dashed line, and a maximum VAAR of 96.15%, marked by the green dotted line. The histogram suggests a bimodal distribution, with peaks around 50% and 70% VAAR values.

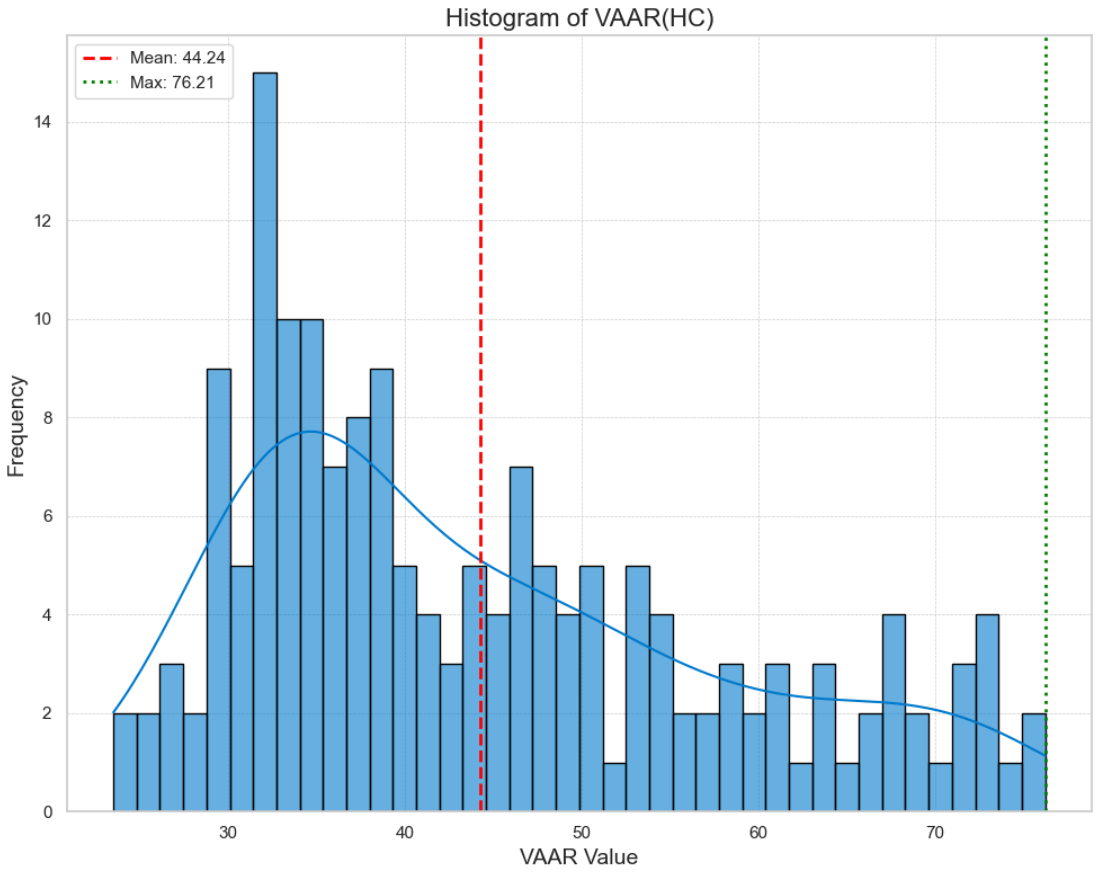

Figure 4.6 Histogram of VAAR (%) for Heavy Chain (HC)

This histogram illustrates the distribution of VAAR values across heavy chain sequences. A mean VAAR of 44.24% is denoted by a red dashed line, while the maximum value observed is 76.21%, shown by the green dotted line. The distribution is somewhat skewed towards lower VAAR values, with a significant peak around 30%-40%.

4.4.3. Sequence Identity

For the test set, the sequence identity for light chain sequences between the original and predicted were recorded at the average of ~76% and maximum average of ~97%. The mean sequence identity of ~60% and maximum of ~84% for predicted heavy chain sequences were recorded. SeqID distribution across the test set is represented in the histogram below for both the chain sequences.

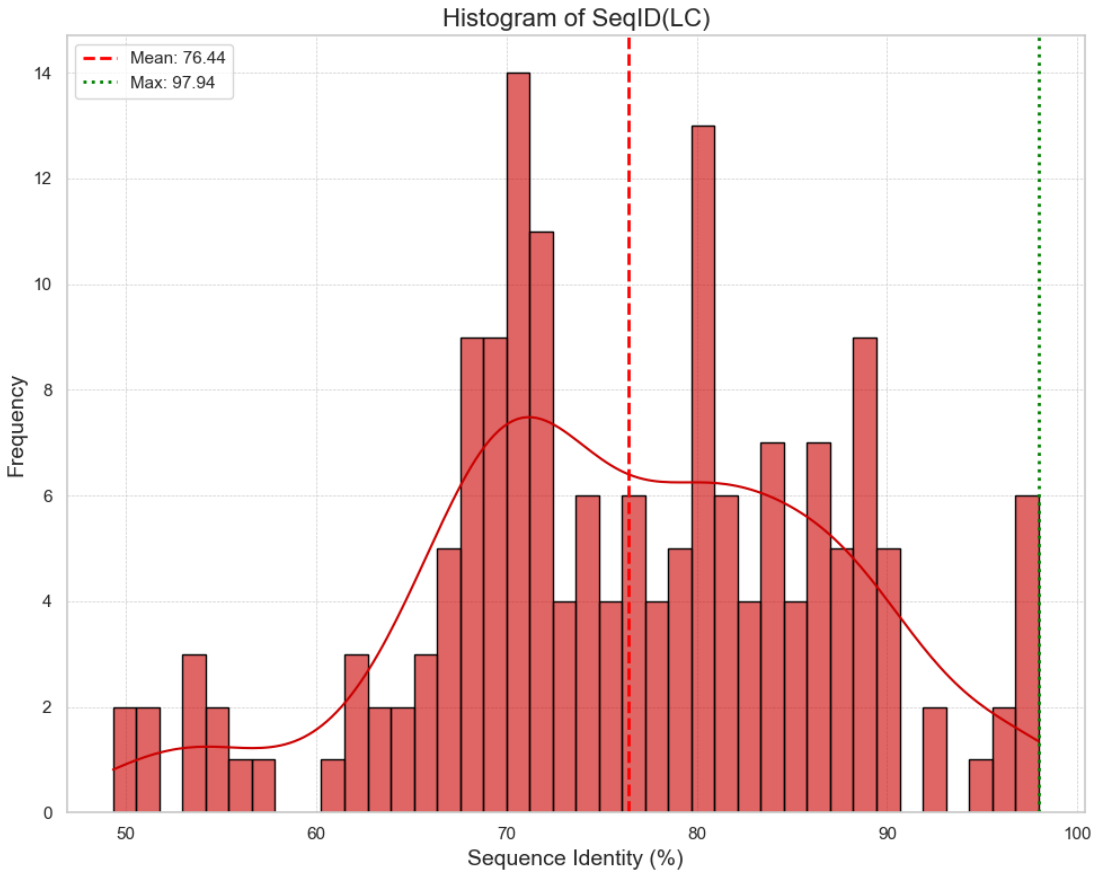

Figure 4.7 Histogram of Sequence Identity (SeqID) for Light Chains (LC).

This histogram presents the distribution of sequence identity percentages for light chain sequences, with a mean SeqID of 76.44% depicted by the red dashed line, and the highest recorded SeqID at 97.94%, marked by the green dotted line.

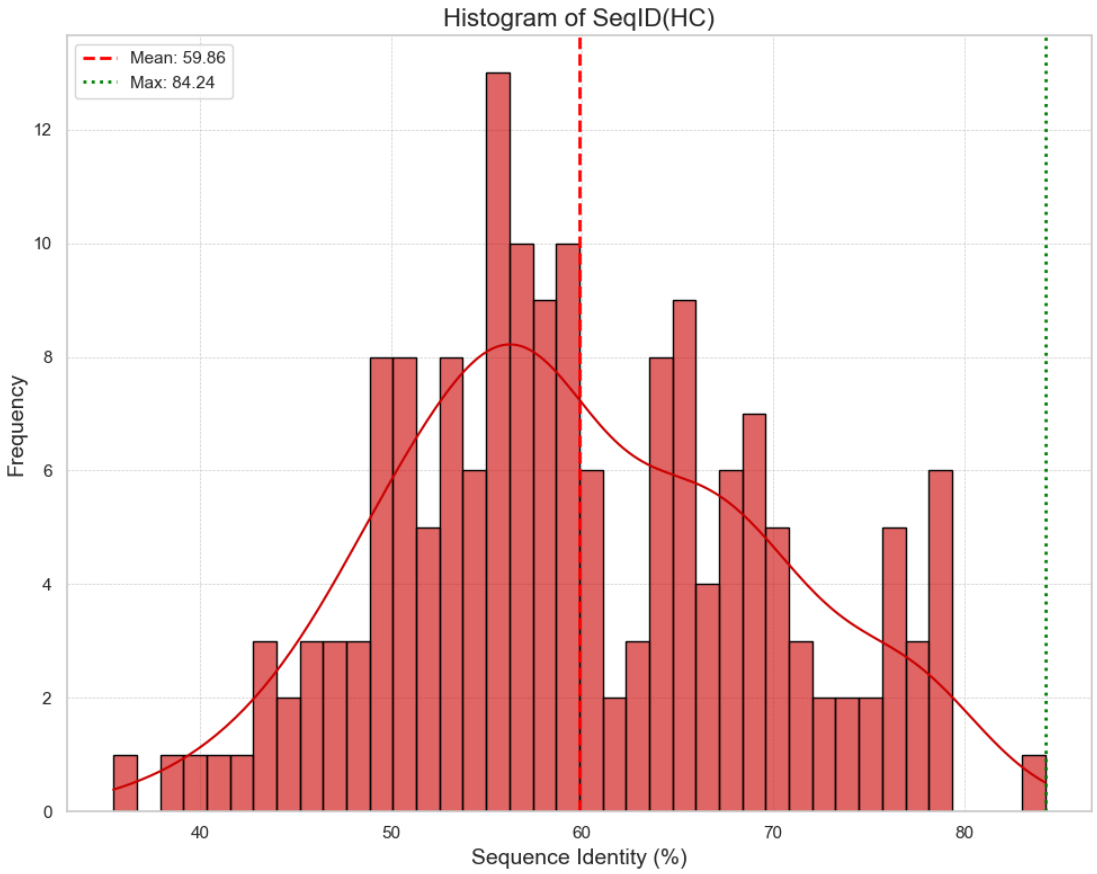

Figure 4.8 Histogram of Sequence Identity (SeqID) for Heavy Chains (HC).

The distribution of sequence identity for heavy chains is showcased in this histogram, with the mean SeqID value of 59.86% indicated by the red dashed line and the maximum SeqID observed at 84.24%, shown by the green dotted line.

The above results are concluded with the help of a boxplot (Figure 4.9 & 4.10) as well as giving a brief summarization of the distribution of evaluation metrics for the test set. For heavy chains, the median VAAR is around 40.33, with the data ranging from a minimum of approximately 23.49 to a maximum of around 76.21. This suggests that while most heavy chain sequences show moderate variability, some sequences exhibit significantly higher or lower variability. The SeqID for heavy chains has a narrower distribution, centering around a median of 58.36, with less spread in the data, indicating a more consistent level of sequence identity across samples. The Recovery Rate for heavy chains shows a mean value near 38.15, with a widespread, which suggests varying levels of recovery efficiency among different sequences.

In contrast, light chains demonstrate higher median values across all three metrics, with a median VAAR of approximately 46.24, a median SeqID of around

76.42, and a particularly high median Recovery Rate of 66.19. The wider IQR in light chains, especially noticeable in the Recovery Rate, highlights a broader distribution of outcomes, with some reaching as high as 97.29 in recovery efficiency. The higher median and maximum values observed in light chains for SeqID and Recovery Rate suggest a generally more effective and consistent antigen-binding capability compared to heavy chains.

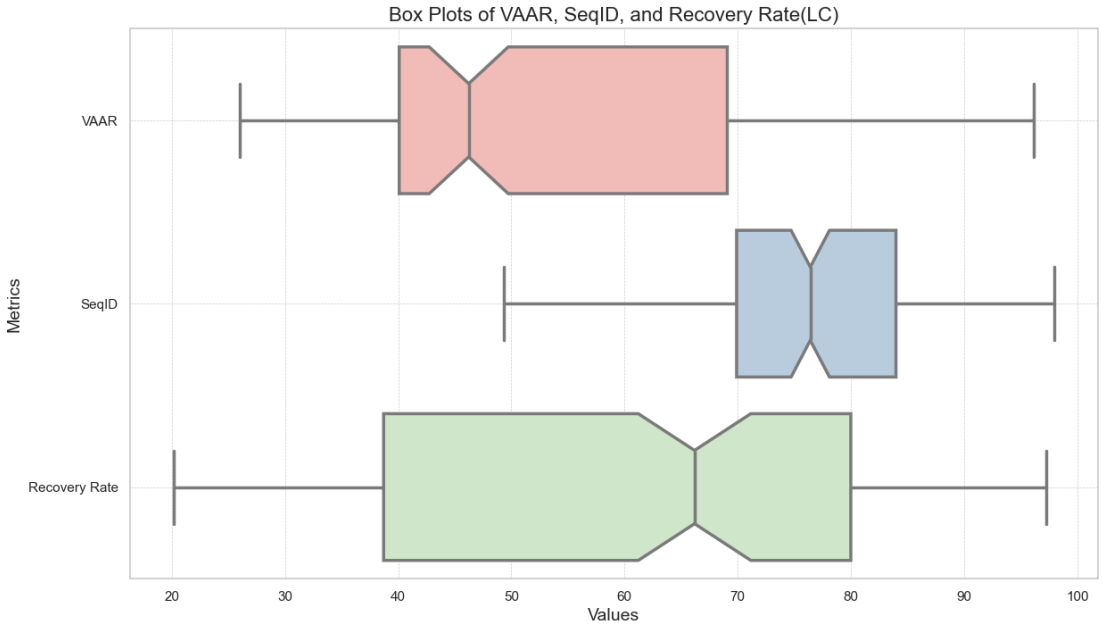

Figure 4.9 Box Plots of VAAR, SeqID, and Recovery Rate for Light Chains (LC)

This figure displays box plots summarizing the distribution of VAAR, sequence identity (SeqID), and recovery rate for light chain sequences. The plots highlight the median, interquartile range, and outliers for each metric.

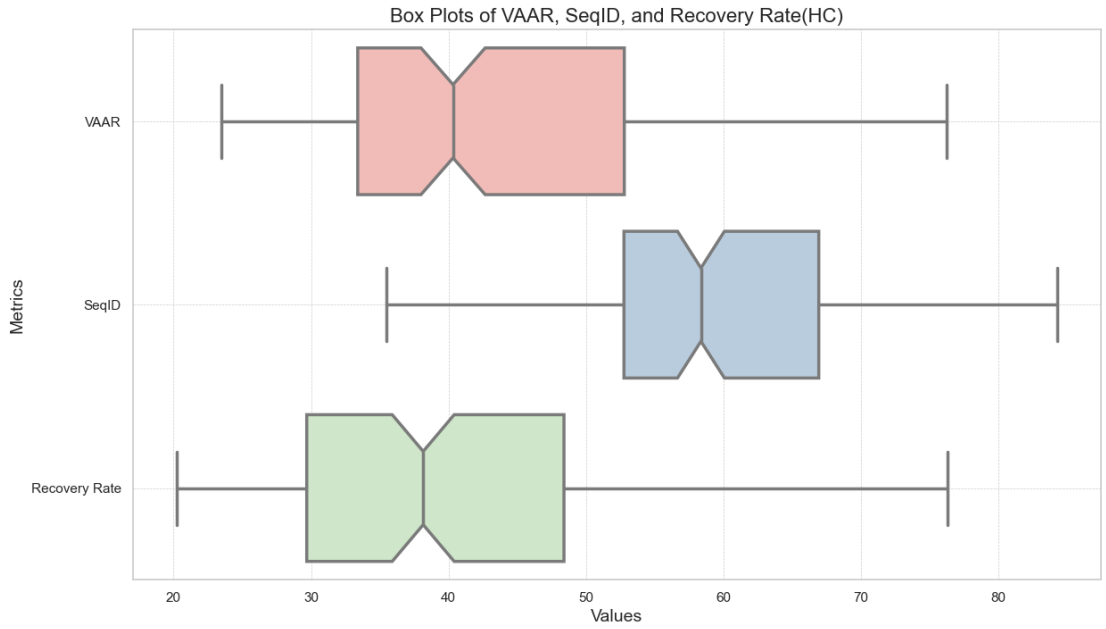

Figure 4.10 Box Plots of VAAR, SeqID, and Recovery Rate for Heavy Chains (HC).

This figure illustrates the distribution characteristics of VAAR, SeqID, and recovery rate across heavy chain sequences through box plots.

4.5.Comparison with related work

This study used the full-length sequences of the Fv region of light and heavy chains to finetune the models along with incorporating the antigen sequence. While previous reported methods predict only the CDRs of heavy and light chains. To compare this study with the previous work, CDRs were determined and then AAR was calculated for CDRs. Previous studies also used the AAR, VAAR, and SeqID for comparison with other related works. For this purpose, 10 complexes from the test set were selected, and their complementarity-determining regions (CDRs) were identified using an online tool, Antibody CDR Annotation, available at Novoprolabs (https://www.novoprolabs.com/tools/cdr). The annotation for original antibody sequences for the selected 10 complexes was also done using the same tool and then the evaluation was performed. The following table provides a comparison of reported values from this study. This comparison is reported by Cohen *et al*, in their study. They evaluated their model with HERN on the test data. Their test data comprises those complexes from SabDAb that became part of the database after time they extracted data from SabDab, meaning the data of the test set was unseen. Building on this foundation, the model was evaluated using completely unseen data. This approach was employed

in the current study, and the results were compared with those reported by Cohen et al. The following table presents a comparison between this study, EAGLE, and HERN for the CDR3 of the heavy chain only.

Table 4.2 Comparison of this study with previous studies.

	VAAR%		SeqID%	
Method	Max	Mean	Max	Mean
HERN	51.96	40.48	51.96	40.48
EAGLE	46.4	34.53	60.93	45.18
AbAtT5	**53.84**	**41.11**	**70.00**	**51.82**

CHAPTER 5: DISCUSSION

Antibodies act as the first reaction forces against the antigens particularly neutralizing the attack by binding with them specifically. Due to their wide application, potential lead candidates of antibodies for a specific purpose are developed in laboratories. These candidates are the results of a wide range of techniques which are laborious, time, and resources. Besides all these laborious works, the lead potential candidates are not promising these technologies. To overcome this, different computational methods have been developed to give libraries leading potential candidates. With the advancements of generative AI and their promising applications in antibody development, researchers started to build libraries of candidates that take fewer resources. These generative models use typical NLP-based architectures, pre-trained on antibody sequences, and generate antibodies. There are mainly two types of these antibody models, which generate antibodies without incorporating antigen sequences. For example, IgLM and Ankh, models generate antibodies that are very close to natural antibodies, and their generated antibodies have been validated. While other models incorporate antigen sequential or structural data to generate epitope-specific antibodies. But these models are limited to short sequences of antigen i.e. only epitope sequence and generate only CDR regions of antibodies. This study uses full-length antigens to generate antibodies against them. This study propped a model, **AbAtT5**, which generates antigen-specific antibodies and showed promising results. This model is a transformer-based encoder-decoder model finetuned on the SabDab dataset. Behind this fine-tuned model, a large protein model, ProtT5 is working, which was trained on a large corpus of protein data from UniRef and BFD.

This study demonstrated the ability of a finetuned model on a specific task and took this task as Neural Machine Translation (NMT), where antigens are given to the model as input and antibody light and heavy chain sequences are taken as the output. The generated sequences are then evaluated using the AAR, VAAR, and SeqID and compared the results with the previous related model. The test data for these evaluations were also taken from SabDab. The study by Cohen *et al* proposed their model and compared it to the HERN model on VAAR and SeqID. The test data in their study was also taken from SabDab, this data set was unseen, and the data was retrieved from SabDab which was published in the database after the date for training data. The same

approach was also used for this study and 170 such complexes were retrieved that were published in the SabDab database after the acquisition of training data. For these test datasets, 10 antibodies were generated against each antigen resulting in a total of 1700 predicted antibody light and heavy chain sequences. The average AAR for light chain sequences was 60.08% while for heavy chain sequences, it was ~40%. The VAAR and SeqID of CDRH3 were compared to the previous study for the top 10 results. This study demonstrated that AbAtT5 outperformed both HERN and EAGLE across two key metrics. For VAAR, AbAtT5 showed a maximal improvement of 1.88% over HERN and 7.44% over EAGLE, with average VAAR increases of 0.63% and 6.58%, respectively. In terms of SeqID, AbAtT5 surpassed HERN by 18.04% in maximal SeqID and 11.34% in average SeqID, while outperforming EAGLE by 9.07% in maximal SeqID and 6.64% in average SeqID.

It is seen that this study has achieved better results than the previously reported work to harness the potential of this study to further extend by incorporating more data. Still, improvement can be made to propose more accurate models for CDRH3 in the future and this will lead to a more potential antigen-specific lead candidate of antibodies.

CHAPTER 6: CONCLUSION

This study introduced AbAtT5, a finetuned transformer-based model, demonstrating its effectiveness in generating antigen-specific antibodies by using full-length antigen sequences. Compared to previous models like HERN and EAGLE, AbAtT5 showed superior performance, with improvements of up to 1.88% in VAAR and 18.04% in SeqID. These results highlight the potential of AbAtT5 in accelerating the antibody design process by offering higher accuracy in sequence generation.

The ability of AbAtT5 to generate antibodies with superior sequence identity and alignment rates underscores its potential as a powerful tool in antibody design. This advancement in computational antibody generation can significantly reduce the time and resources required in traditional laboratory methods, offering a more efficient pathway to identifying potent antigen-specific antibody candidates.

Future work will aim to enhance the accuracy of models like AbAtT5 by incorporating additional data and refining the model architecture, particularly in the generation of CDRH3 regions. These efforts will further improve the precision of antigen-specific antibody prediction, leading to the development of more effective therapeutic candidates.

REFERENCES

[1] R. H. Khan and P. Salahuddin, “Protein Structure and Function,” 2017, Accessed: Aug. 27, 2024. [Online]. Available: www.austinpublishinggroup.com/ebooks

[2] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, “Protein Function,” 2002, Accessed: Aug. 27, 2024. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK26911/

[3] S. R. Shenoy and B. Jayaram, “Proteins: Sequence to Structure and Function - Current Status,” *Curr Protein Pept Sci*, vol. 11, no. 7, pp. 498–514, Jan. 2011, doi: 10.2174/138920310794109094.

[4] Y. Choi, G. E. Sims, S. Murphy, J. R. Miller, and A. P. Chan, “Predicting the Functional Effect of Amino Acid Substitutions and Indels,” *PLoS One*, vol. 7, no. 10, p. e46688, Oct. 2012, doi: 10.1371/JOURNAL.PONE.0046688.

[5] M. Suhail, “Biophysical chemistry behind sickle cell anemia and the mechanism of voxelotor action,” *Scientific Reports 2024 14:1*, vol. 14, no. 1, pp. 1–18, Jan. 2024, doi: 10.1038/s41598-024-52476-8.

[6] M. S. Maddur, S. Lacroix-Desmazes, J. D. Dimitrov, M. D. Kazatchkine, J. Bayry, and S. V. Kaveri, “Natural Antibodies: from First-Line Defense Against Pathogens to Perpetual Immune Homeostasis,” *Clinical Reviews in Allergy & Immunology 2019 58:2*, vol. 58, no. 2, pp. 213–228, Jun. 2019, doi: 10.1007/S12016-019-08746-9.

[7] Lauren. Sompayrac, “How the immune system works,” p. 163, 2023, Accessed: Aug. 27, 2024. [Online]. Available: https://books.google.com/books/about/How_the_Immune_System_Works.html?id=lVKMEAAAQBAJ

[8] B. A. Keyt, R. Baliga, A. M. Sinclair, S. F. Carroll, and M. S. Peterson, “Structure, Function, and Therapeutic Use of IgM Antibodies,” *Antibodies 2020, Vol. 9, Page 53*, vol. 9, no. 4, p. 53, Oct. 2020, doi: 10.3390/ANTIB9040053.

[9] R. A. Norman *et al.*, “Computational approaches to therapeutic antibody design: established methods and emerging trends,” *Brief Bioinform*, vol. 21, no. 5, pp. 1549–1567, doi: 10.1093/bib/bbz095.

[10] T. Te Wu and E. A. Kabat, “AN ANALYSIS OF THE SEQUENCES OF THE VARIABLE REGIONS OF BENCE JONES PROTEINS AND MYELOMA LIGHT CHAINS AND THEIR IMPLICATIONS FOR ANTIBODY COMPLEMENTARITY,” *J Exp Med*, vol. 132, no. 2, p. 211, Aug. 1970, doi: 10.1084/JEM.132.2.211.

[11] C. Chothia and A. M. Lesk, “Canonical structures for the hypervariable regions of immunoglobulins,” *J Mol Biol*, vol. 196, no. 4, pp. 901–917, Aug. 1987, doi: 10.1016/0022-2836(87)90412-8.

[12] M. P. Lefranc, “IMGT Unique Numbering for the Variable (V), Constant (C), and Groove (G) Domains of IG, TR, MH, IgSF, and MhSF,” *Cold Spring Harb Protoc*, vol. 2011, no. 6, p. pdb.ip85, Jun. 2011, doi: 10.1101/PDB.IP85.

[13] B. B. Haab, “Applications of antibody array platforms,” *Curr Opin Biotechnol*, vol. 17, no. 4, pp. 415–421, Aug. 2006, doi: 10.1016/J.COPBIO.2006.06.013.

[14] J. Ma *et al.*, “Bispecific Antibodies: From Research to Clinical Application,” *Front Immunol*, vol. 12, p. 626616, May 2021, doi: 10.3389/FIMMU.2021.626616/BIBTEX.

[15] E. D. G. Fleuren *et al.*, “Theranostic applications of antibodies in oncology,” *Mol Oncol*, vol. 8, no. 4, pp. 799–812, Jun. 2014, doi: 10.1016/J.MOLONC.2014.03.010.

[16] A. J. Killard, B. Deasy, R. O’Kennedy, and M. R. Smyth, “Antibodies: production, functions and applications in biosensors,” *TrAC Trends in Analytical Chemistry*, vol. 14, no. 6, pp. 257–266, Jun. 1995, doi: 10.1016/0165-9936(95)91618-3.

[17] H. F. Jones, Z. Molvi, M. G. Klatt, T. Dao, and D. A. Scheinberg, “Empirical and Rational Design of T Cell Receptor-Based Immunotherapies,” *Front Immunol*, vol. 11, p. 585385, Jan. 2021, doi: 10.3389/FIMMU.2020.585385/BIBTEX.

[18] H. A. Parray *et al.*, “Hybridoma technology a versatile method for isolation of monoclonal antibodies, its applicability across species, limitations, advancement and future perspectives,” *Int Immunopharmacol*, vol. 85, p. 106639, Aug. 2020, doi: 10.1016/J.INTIMP.2020.106639.

[19] L. Ledsgaard *et al.*, “Advances in antibody phage display technology,” *Drug Discov Today*, vol. 27, no. 8, pp. 2151–2169, Aug. 2022, doi: 10.1016/J.DRUDIS.2022.05.002.

[20] B. Samuel Schmitz, J. Meiler, J. E. Crowe Jr, L. Plate, and L. Buchanan, “Computational methods to engineer antibodies for vaccines and therapeutics,” 2022.

[21] F. Barreto, L. Moharkar, M. Shirodkar, V. Sarode, S. Gonsalves, and A. Johns, “Generative Artificial Intelligence: Opportunities and Challenges of Large Language Models,” *Lecture Notes in Networks and Systems*, vol. 699 LNNS, pp. 545–553, 2023, doi: 10.1007/978-981-99-3177-4_41.

[22] A. Aggarwal, M. Mittal, and G. Battineni, “Generative adversarial network: An overview of theory and applications,” *International Journal of Information Management Data Insights*, vol. 1, no. 1, p. 100004, Apr. 2021, doi: 10.1016/J.JJIMEI.2020.100004.

[23] A. Vaswani *et al.*, “Attention Is All You Need”.

[24] N. Patwardhan, S. Marrone, and C. Sansone, “Transformers in the Real World: A Survey on NLP Applications,” *Information 2023, Vol. 14, Page 242*, vol. 14, no. 4, p. 242, Apr. 2023, doi: 10.3390/INFO14040242.

[25] L. Banh and G. Strobel, “Generative artificial intelligence,” *Electronic Markets 2023 33:1*, vol. 33, no. 1, pp. 1–17, Dec. 2023, doi: 10.1007/S12525-023-00680-1.

[26] A. Bandi, P. V. S. R. Adapa, and Y. E. V. P. K. Kuchi, “The Power of Generative AI: A Review of Requirements, Models, Input–Output Formats, Evaluation Metrics, and

Challenges," *Future Internet 2023, Vol. 15, Page 260*, vol. 15, no. 8, p. 260, Jul. 2023, doi: 10.3390/FI15080260.

[27] A. Strokach and P. M. Kim, "Deep generative modeling for protein design," *Curr Opin Struct Biol*, vol. 72, pp. 226–236, Feb. 2022, doi: 10.1016/J.SBI.2021.11.008.

[28] Y. Cao, T. A. Geddes, J. Y. H. Yang, and P. Yang, "Ensemble deep learning in bioinformatics," *Nature Machine Intelligence 2020 2:9*, vol. 2, no. 9, pp. 500–508, Aug. 2020, doi: 10.1038/s42256-020-0217-y.

[29] J. Jumper *et al.*, "Highly accurate protein structure prediction with AlphaFold," *Nature 2021 596:7873*, vol. 596, no. 7873, pp. 583–589, Jul. 2021, doi: 10.1038/s41586-021-03819-2.

[30] Y. Lei *et al.*, "A deep-learning framework for multi-level peptide–protein interaction prediction," *Nature Communications 2021 12:1*, vol. 12, no. 1, pp. 1–10, Sep. 2021, doi: 10.1038/s41467-021-25772-4.

[31] F. Zhang, Y. Zhang, X. Zhu, X. Chen, F. Lu, and X. Zhang, "DeepSG2PPI: A Protein-Protein Interaction Prediction Method Based on Deep Learning," *IEEE/ACM Trans Comput Biol Bioinform*, vol. 20, no. 5, pp. 2907–2919, Sep. 2023, doi: 10.1109/TCBB.2023.3268661.

[32] T. Yun, H. Li, P. C. Chang, M. F. Lin, A. Carroll, and C. Y. McLean, "Accurate, scalable cohort variant calls using DeepVariant and GLnexus," *Bioinformatics*, vol. 36, no. 24, pp. 5582–5589, Apr. 2021, doi: 10.1093/BIOINFORMATICS/BTAA1081.

[33] N. Sapoval *et al.*, "Current progress and open challenges for applying deep learning across the biosciences," *Nature Communications 2022 13:1*, vol. 13, no. 1, pp. 1–12, Apr. 2022, doi: 10.1038/s41467-022-29268-7.

[34] T. Wolf *et al.*, "Transformers: State-of-the-Art Natural Language Processing," *EMNLP 2020 - Conference on Empirical Methods in Natural Language Processing, Proceedings of Systems Demonstrations*, pp. 38–45, 2020, doi: 10.18653/V1/2020.EMNLP-DEMOS.6.

[35] J. Wang, H. Cao, J. Z. H. Zhang, and Y. Qi, "Computational Protein Design with Deep Learning Neural Networks," *Scientific Reports 2018 8:1*, vol. 8, no. 1, pp. 1–9, Apr. 2018, doi: 10.1038/s41598-018-24760-x.

[36] N. Ferruz, S. Schmidt, and B. Höcker, "ProtGPT2 is a deep unsupervised language model for protein design," *Nature Communications 2022 13:1*, vol. 13, no. 1, pp. 1–10, Jul. 2022, doi: 10.1038/s41467-022-32007-7.

[37] E. Castro, A. Godavarthi, J. Rubinfien, K. Givechian, D. Bhaskar, and S. Krishnaswamy, "Transformer-based protein generation with regularized latent space optimization," *Nature Machine Intelligence 2022 4:10*, vol. 4, no. 10, pp. 840–851, Sep. 2022, doi: 10.1038/s42256-022-00532-1.

[38] J. E. Shin *et al.*, "Protein design and variant prediction using autoregressive generative models," *Nature Communications 2021 12:1*, vol. 12, no. 1, pp. 1–11, Apr. 2021, doi: 10.1038/s41467-021-22732-w.

[39] R. W. Shuai, J. A. Ruffolo, and J. J. Gray, “IgLM: Infilling language modeling for antibody sequence design,” *Cell Syst*, vol. 14, no. 11, pp. 979-989.e4, Nov. 2023, doi: 10.1016/j.cels.2023.10.001.

[40] D. M. Mason *et al.*, “Optimization of therapeutic antibodies by predicting antigen specificity from antibody sequence via deep learning,” *Nat Biomed Eng*, vol. 5, no. 6, pp. 600–612, Jun. 2021, doi: 10.1038/s41551-021-00699-9.

[41] J. Liu, M. Yang, Y. Yu, H. Xu, K. Li, and X. Zhou, “Large language models in bioinformatics: applications and perspectives,” *ArXiv*, Jan. 2024, Accessed: Aug. 27, 2024. [Online]. Available: /pmc/articles/PMC10802675/

[42] N. Q. K. Le, “Leveraging transformers-based language models in proteome bioinformatics,” *Proteomics*, vol. 23, no. 23–24, p. 2300011, Dec. 2023, doi: 10.1002/PMIC.202300011.

[43] K. Saka *et al.*, “Antibody design using LSTM based deep generative model from phage display library for affinity maturation,” *Scientific Reports 2021 11:1*, vol. 11, no. 1, pp. 1–13, Mar. 2021, doi: 10.1038/s41598-021-85274-7.

[44] T. Cohen and D. Schneidman-Duhovny, “Epitope-specific antibody design using diffusion models on the latent space of ESM embeddings.”

[45] K. Gao *et al.*, “Pre-training Antibody Language Models for Antigen-Specific Computational Antibody Design,” *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 23, pp. 506–517, Aug. 2023, doi: 10.1145/3580305.3599468/SUPPL_FILE/RTFP1413-2MIN-PROMO.MP4.

[46] J. Dunbar *et al.*, “SAbDab: the structural antibody database,” *Nucleic Acids Res*, vol. 42, no. D1, pp. D1140–D1146, Jan. 2014, doi: 10.1093/NAR/GKT1043.

[47] A. Elnaggar *et al.*, “ProtTrans: Toward Understanding the Language of Life Through Self-Supervised Learning,” *IEEE Trans Pattern Anal Mach Intell*, vol. 44, no. 10, pp. 7112–7127, Oct. 2022, doi: 10.1109/TPAMI.2021.3095381.