[2022 Summer/Fall] KAIST URP Program

# Individual Research Project Final Report

[Research Participation Project / Creative Project]

# 병원성 바이러스에 대응하는 항체 설계를 위한 대조적 학습

# Contrastive Learning for antibody representation learning and antibody classification targeting pathogenic viruses

Department – School of Computing, Anar Rzayev (20190788)

# For Submission

**Research Subject (Korean): 병원성 바이러스에 대응하는 항체 설계를 위한 대조적 학습**

**Research Subject (English): Contrastive learning for antibody representation learning and antibody classification targeting pathogenic viruses**

**Research Period : June 27, 2022 ~ December 16, 2022**

**Advisory Professor : Ho Min Kim (Signature)**

**Teaching Assistant : Hyunkyu Jung** 정현규 **(Signature)**

**As a participant(s) of the KAIST URP program, I (We) have completed the above research and hereby submit the final report on the research.**

**December 23, 2022**

**Researcher Anar Rzayev (Signature)**

# Contents

# Abstract

Learning effective antibody representations is fundamental in multiple biological tasks, specifically for predicting antigen specificity to target pathogenic viruses. Current methodologies usually pretrain language models on a large number of unlabeled amino acid sequences of antibodies and then hyper-tune the models with some labeled data in downstream applications. Even though the sequence-based methods have proven to be effective in multiple realms, the power of pretraining on known antibody structures, which are available in exponentially small magnitude, has not been explored for antigen specificity predictions.

In this project, we will explore a simple yet effective structure-based encoder for antibody representation learning to embed the geometric features of antibodies according to their 3D structural information. We pretrain the antibody graph encoder by leveraging Multiview contrastive learning and evaluate the antigen-specificity task for Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2), Hemagglutinin Influenza (HA), and Human Immunodeficiency Virus (HIV). Experimental results on those viruses indicate comparatively high values for precision/recall/F1-score and reveal qualitatively accurate visualization of latent space for 20 protein families.

## 1. Research Background

In this section, we explore the biological background of our research, its related previous works on traditional computational methods, and recent approaches applied with Deep Learning.

### Background

The adaptive immune system of vertebrates is capable of mounting robust responses to a broad range of potential pathogens. Critical to this flexibility are antibodies, which are specialized to recognize a diverse set of molecular patterns with high affinity and specificity. The overall role of an antibody is to bind to an antigen, e.g., a virus, present it to the immune system, and stimulate an immune response. This natural role in the defense against pathogens, e.g. SARS-COV-2, Influenza, makes antibodies an increasingly popular choice for the development of new therapeutics.

An antibody consists of a heavy chain and a light chain, each composed of a variable domain (VH/VL) and a constant domain, as shown in Fig. 1.1. The variable domain is further divided into a framework region and three complementarity-determining regions (CDRs). The three CDRs on the heavy chain are denoted as CDR-H1, CDR-H2, CDRH3, each occupying a contiguous subsequence in the framework region sequence. As the most variable part of an antibody, CDRs are the main determinants of binding and neutralization. Following current state-of-art approaches in computational biology [4-9], we formulate antibody design as a CDR generation problem, conditioned on the framework region (Fab) sequence.

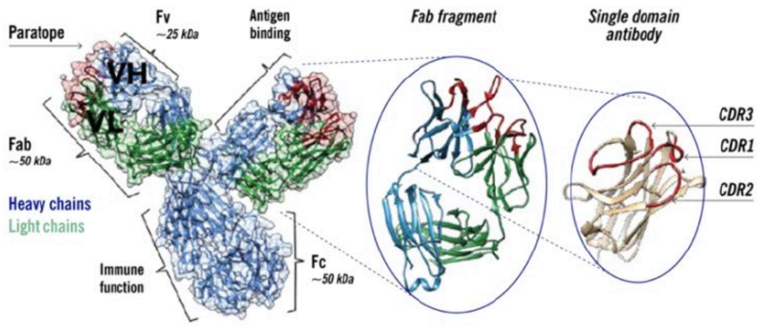


**Figure 1.1**: Structures of an antibody and of common antibody fragments. An antibody structure is shown on the left with the heavy (H) and light (L) chains in blue and green, respectively. CDRs containing the paratopes are colored in red, and the heavy and light variable domains (VH and VL) are labelled. The antigen-binding fragment (Fab) region is responsible for recognizing the target, while the crystallizable fragment (Fc) region for immune function and lysosome escape. The three CDR loops are highlighted in red on VH domain.

Currently, monoclonal antibodies make up a rapidly growing segment of the global pharmaceutical market. The global therapeutic monoclonal antibody market was valued at approximately $150 billion in 2019 and is expected to generate revenue of $300 billion by the end of 2025 [13]. However, rational design of antibody-antigen interactions is hindered by reliance on experimental methods such as crystallography, NMR, and cryo-EM, which are low throughput and requires significant investments of time and resources that may fail (Fig. 1.2).

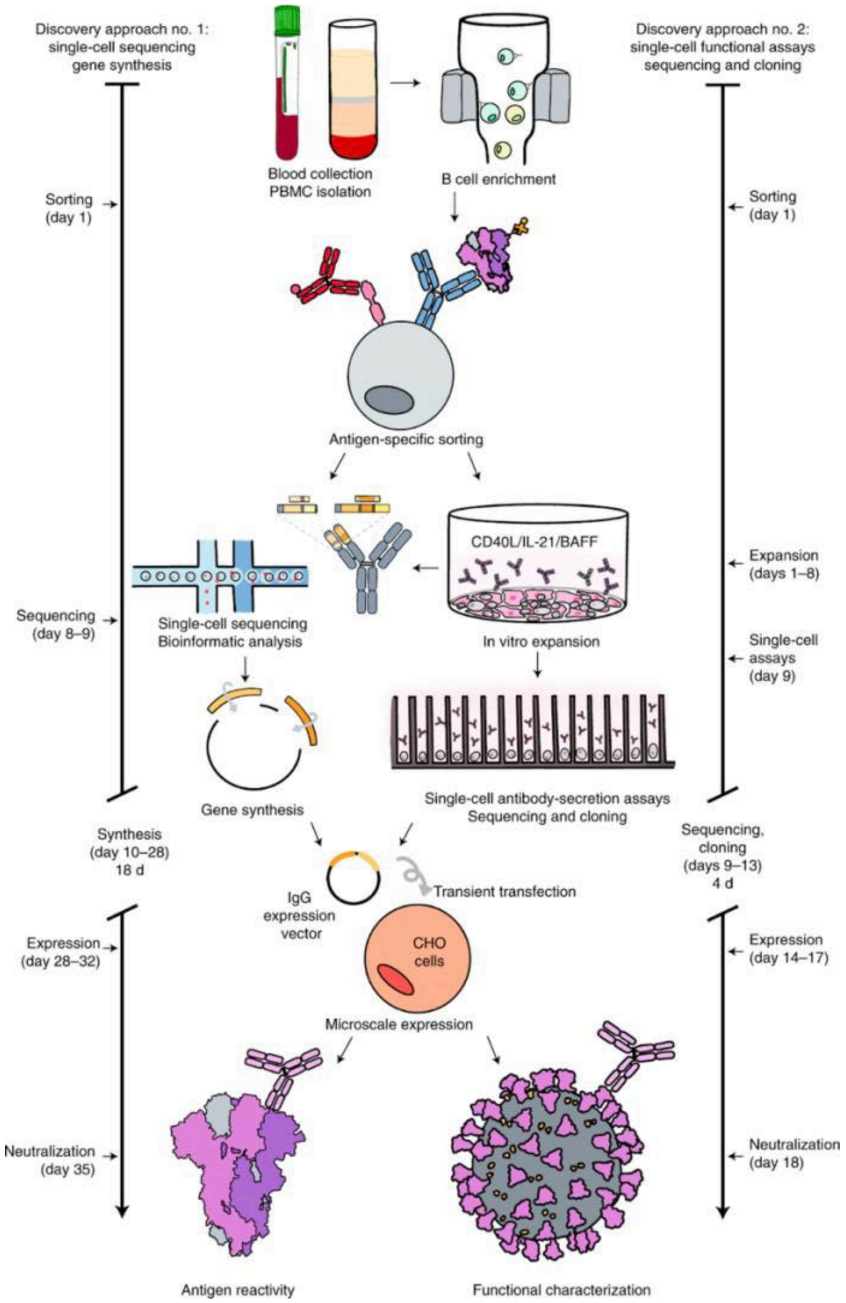

Discovery approach no. 1: single-cell sequencing gene synthesis
Discovery approach no. 2: single-cell functional assays sequencing and cloning
Blood collection PBMC isolation
B cell enrichment
Sorting (day 1)
Sorting (day 1)
Antigen-specific sorting
CD40L/IL-21/BAFF
Expansion (days 1–8)
Sequencing (day 8–9)
Single-cell sequencing Bioinformatic analysis
In vitro expansion
Single-cell assays (day 9)
Gene synthesis
Single-cell antibody-secretion assays Sequencing and cloning
Synthesis (day 10–28) 18 d
Sequencing, cloning (days 9–13) 4 d
IgG expression vector
Transient transfection
CHO cells
Expression (day 28–32)
Expression (day 14–17)
Microscale expression
Neutralization (day 35)
Neutralization (day 18)
Antigen reactivity
Functional characterization

**Figure 1.2**: Rational design of antibody-antigen interactions

### Traditional Methods

In general, methods for computational antibody design roughly fall into two categories. The first class is based on energy function optimization, which uses Markov Chain Monte Carlo simulation to iteratively modify an antibody sequence and its structure to reach a local minimum energy for the antibody structure and the interface between antibody and antigen (Fig. 1.3). Similar approaches are also used in protein design [3, 4]. However, these physics-based methods are computationally expensive, in which the designed sequence can fold into a structure different from the designed structure, and our antigen-conditioned objective can be more complicated than evaluating only physics-based binding energy models [5, 9].

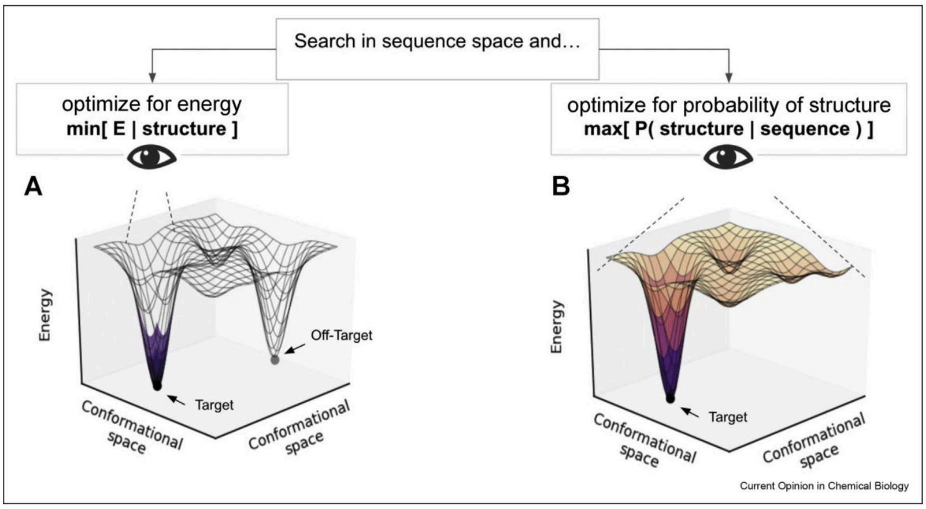
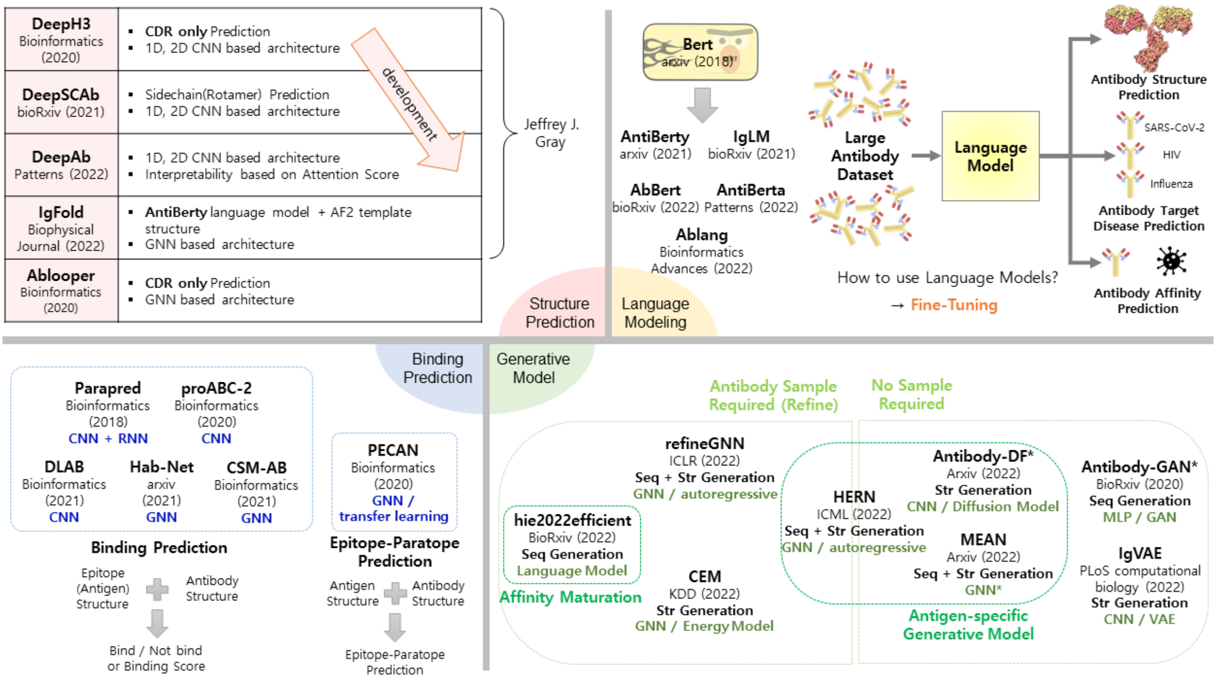


**Figure 1.3**: Energy Optimization of Protein Sequences. [Christoffer Norn et al. 2021]

### Deep Learning-based Methods

The second methodology is based on generative models. For antibodies, they are mostly sequence-based [15, 16], whereas regarding the proteins, authors from [17, 18, 19] further developed models conditioned on a backbone structure or protein folding in general. Since the best CDR structures are often unknown for new pathogens, most approaches codesign sequences and structures for specific properties of targeted viruses, mimicking autoregressive models for graph generation.

In fact, the application of Natural Language Processing (NLP) algorithms, such as Transformers [20], and Graph Neural Networks (GNN) have been proven to be efficient for designing de-novo antibody sequences and structures [6, 7, 8, 9]. Particularly, due to the limited antibody structures available in the structural antibody database [1, 2], most of the recent research studies have focused on antibody sequence generation for paired immunoglobulin sequences using Bidirectional Encoder Representations from Transformers (BERT) models [7, 8, 12] and (self-)supervised learning algorithms [6, 9]. However, there has been a lack of studies that have investigated both structure based and sequence-level representations to filter positive candidates from an antibody dataset which may possibly bind and neutralize specific pathogenic viruses. Figure 1.4 illustrates the overall landscape of AI-based antibody design in terms of Antibody Structure Prediction, Language Modeling, Generative Models, and Binding Predictions. As indicated in the specificity task (i.e. blue), most of such prediction models condition on antigen's epitope structures and utilize simple Convolutional Neural Network (CNN) and/or Graph Neural Networks (GNN) to capture antibody-antigen interactions.



DeepH3 Bioinformatics (2020)
▪ CDR only Prediction
▪ 1D, 2D CNN based architecture
DeepSCAb bioRxiv (2021)
▪ Sidechain(Rotamer) Prediction
▪ 1D, 2D CNN based architecture
DeepAb Patterns (2022)
▪ 1D, 2D CNN based architecture
▪ Interpretability based on Attention Score
IgFold Biophysical Journal (2022)
▪ AntiBerty language model + AF2 template structure
▪ GNN based architecture
Ablooper Bioinformatics (2020)
▪ CDR only Prediction
▪ GNN based architecture
development
Jeffrey J. Gray
Structure Prediction
Language Modeling
Bert arxiv (2018)
AntiBerty arxiv (2021)
IgLM bioRxiv (2021)
AbBert bioRxiv (2022)
AntiBerta Patterns (2022)
Ablang Bioinformatics Advances (2022)
Large Antibody Dataset
Language Model
How to use Language Models?
→ Fine-Tuning
Antibody Structure Prediction
SARS-CoV-2
HIV
Influenza
Antibody Target Disease Prediction
Antibody Affinity Prediction
Binding Prediction
Generative Model
Parapred Bioinformatics (2018) CNN + RNN
proABC-2 Bioinformatics (2020) CNN
DLAB Bioinformatics (2021) CNN
Hab-Net arxiv (2021) GNN
CSM-AB Bioinformatics (2021) GNN
Binding Prediction
Epitope (Antigen) Structure
Antibody Structure
Bind / Not bind or Binding Score
PECAN Bioinformatics (2020) GNN / transfer learning
Epitope-Paratope Prediction
Antigen Structure
Antibody Structure
Epitope-Paratope Prediction
Antibody Sample Required (Refine)
No Sample Required
refineGNN ICLR (2022) Seq + Str Generation GNN / autoregressive
hie2022efficient BioRxiv (2022) Seq Generation Language Model
Affinity Maturation
CEM KDD (2022) Str Generation GNN / Energy Model
HERN ICML (2022) Seq + Str Generation GNN / autoregressive
Antibody-DF* Arxiv (2022) Str Generation CNN / Diffusion Model
MEAN Arxiv (2022) Seq + Str Generation GNN*
Antigen-specific Generative Model
Antibody-GAN* BioRxiv (2020) Seq Generation MLP / GAN
IgVAE PLoS computational biology (2022) Str Generation CNN / VAE

**Figure 1.4**: AI-based antibody development using language modeling and generative models

## 2. Research Purpose

In this work, our aim is to use contrastive learning methods (Fig. 2.1) for learning effective antibody representation (embeddings). In addition, we also intend to propose a proper data augmentation for antibody sequences and their respective structural data.

Contrastive learning has recently achieved good results to classify images in computer vision tasks [10, 11]. Using such self-supervised algorithms, the loss is designed to maximize the difference between positive examples and negative examples. In our study, specifically, positive examples could be regarded as antibodies that bind and neutralize to a specific pathogenic virus, while negative examples would be antibodies which do not bind to the specific antigen. We hypothesize that representation learned by contrastive learning using structural-based and sequence-based data can help in the classification of possible binders for specific antigens and learn good 3D/sequence representations for antigen-specificity tasks.

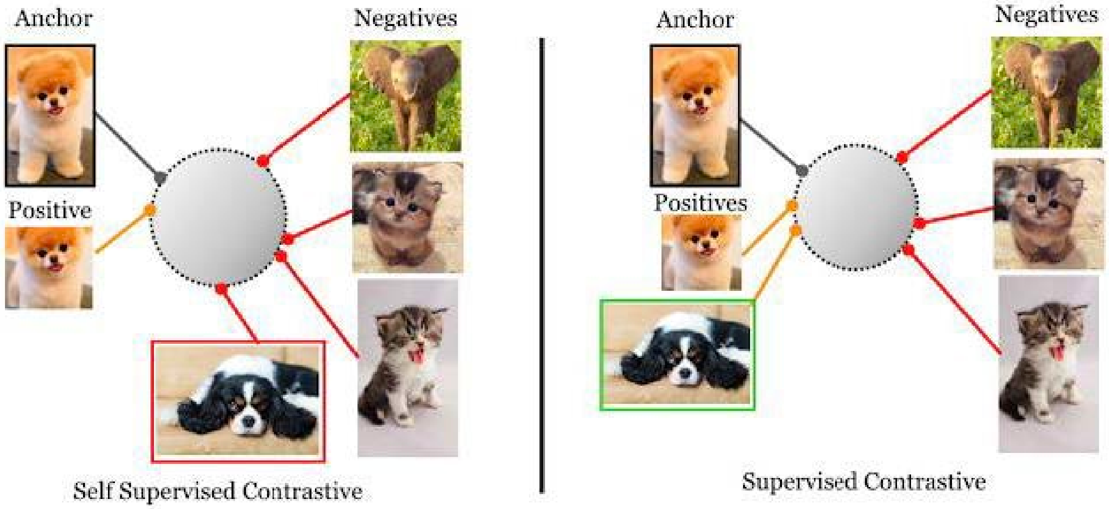


**Figure 2.1**: Supervised vs self-supervised contrastive losses. Supervised contrastive learning considers different samples from the same class as positive examples [link]

Our main objective during this work can be summarized as follows:

- Generate a synthetic antibody structure dataset by using State-Of-The-Art (SOTA) deep learning-based antibody structure prediction networks.
- Train a classifier via contrastive learning to identify if an antibody is a potential binder to a target antigen by using sequence-based and/or structure-based representations (similar to Fig 2.2).

- Propose a novel data augmentation mechanism for antibody data to produce additional synthetic sequences/structures for learning better representations using contrastive learning algorithms.
- Generate novel antigen-specific antibody candidates with traditional computational methods such as Rosetta Antibody Design (RAbD) [3] and deep learning-based generative methods. Evaluate the candidates in silico using the contrastive learning-based classifier, and select the best ones for real wet-lab experiments to compute binding affinity, solubility and developability parameters.

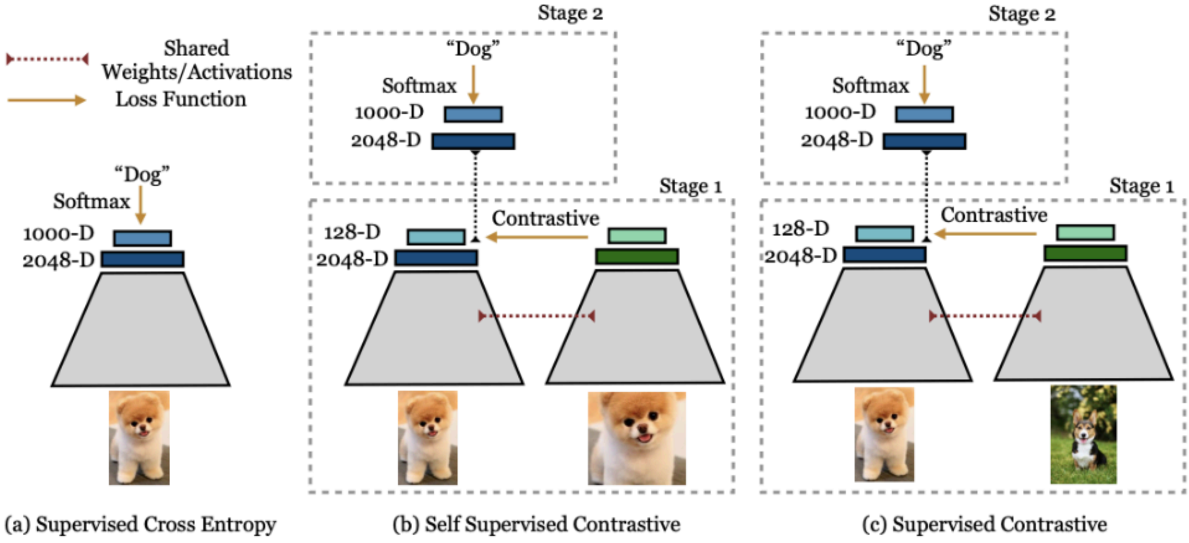
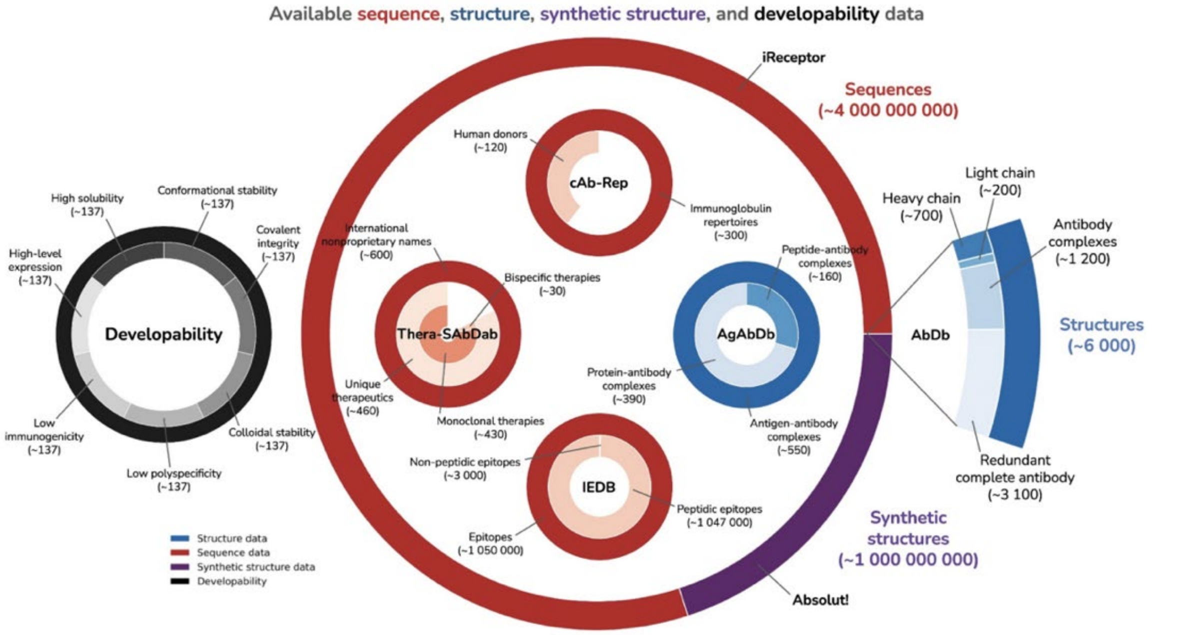


**Figure 2.2**: Cross entropy, self-supervised contrastive loss and supervised contrastive loss: The cross-entropy loss (left) uses labels and a SoftMax loss to train a classifier; the self-supervised contrastive loss (middle) uses a contrastive loss and data augmentations to learn representations. The supervised contrastive loss (right) also learns representations using a contrastive loss, but uses label information to sample positives in addition to augmentations of the same image.

## 3. Preprocessing & Datasets

One of the important aspects of therapeutic antibody design is to collect antigen specific data that could be processed and filtered for efficient learning representations. As we aim to train a (self-)supervised contrastive classifier using both sequence and structure, the first step is to gather those sequence and structure data for antibodies that are confirmed to neutralize target antigens. In the next subsection, we detail how we are collecting the antibody sequences and structures to create a dataset and how we are inferring synthetic structures for antibodies that only contain sequence data using deep learning-based antibody structure prediction networks.

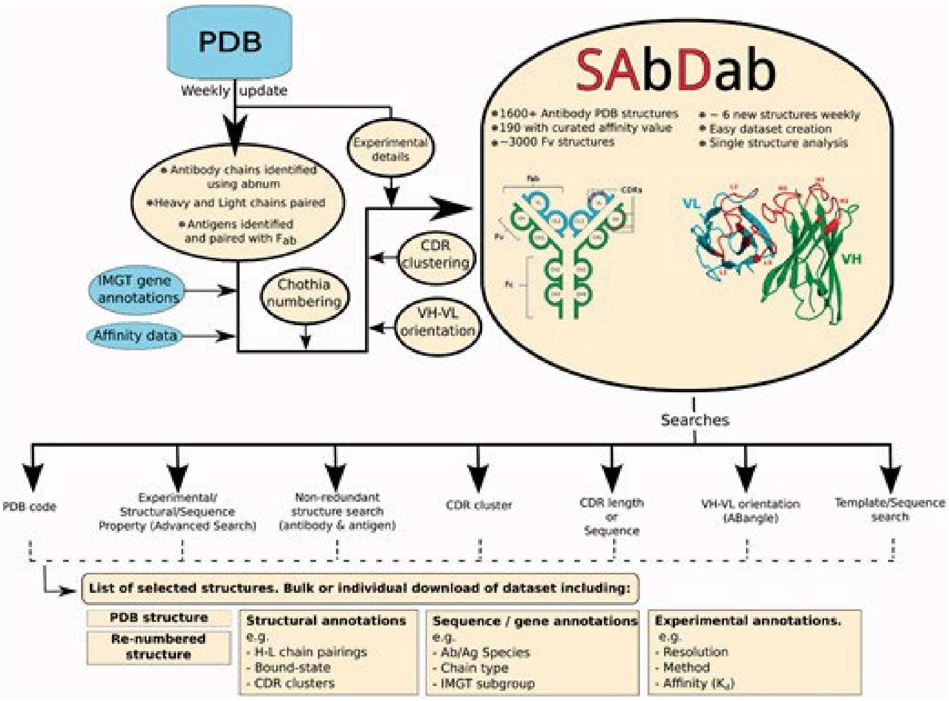


Available sequence, structure, synthetic structure, and developability data
iReceptor
Sequences (~4 000 000 000)
Human donors (~120)
cAb-Rep
Immunoglobulin repertoires (~300)
High solubility (~137)
Conformational stability (~137)
Covalent integrity (~137)
High-level expression (~137)
Developability
Low immunogenicity (~137)
Low polyspecificity (~137)
Colloidal stability (~137)
International nonproprietary names (~600)
Bispecific therapies (~30)
Thera-SAbDab
Unique therapeutics (~460)
Monoclonal therapies (~430)
Peptide-antibody complexes (~160)
AgAbDb
Protein-antibody complexes (~390)
Antigen-antibody complexes (~550)
Non-peptidic epitopes (~3 000)
IEDB
Peptidic epitopes (~1 047 000)
Epitopes (~1 050 000)
Light chain (~200)
Heavy chain (~700)
Antibody complexes (~1 200)
AbDb
Structures (~6 000)
Redundant complete antibody (~3 100)
Synthetic structures (~1 000 000 000)
Absolut!
Structure data
Sequence data
Synthetic structure data
Developability

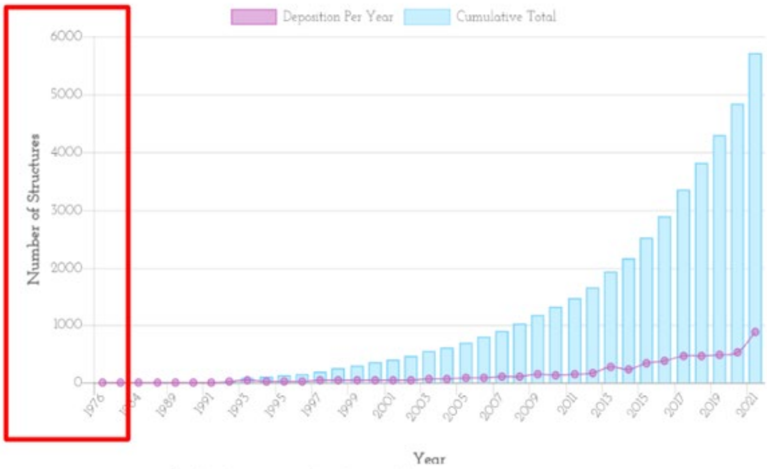
**Figure 3.1.** Available sequence, (synthetic) structure, and developability data for antibodies [link]

### Structural Antibody Database (SAbDab)

Structural antibody database [2] is an online resource containing all the publicly available antibody structures annotated and presented in a consistent fashion. The data are annotated with several properties including experimental information, gene details, correct heavy and light chain pairings, antigen details and, if available, antibody-antigen binding affinity. As in Fig. 3.1, the user can retrieve the full set of structures, specific entries by specifying their Protein Data Bank (PDB) code or to create subsets based on search criteria [1]. Structures can be searched based on the experimental methods used to determine the structure, species of the antibody, antigen type, presence of affinity values in the annotation and presence of amino acid residues at specific sequence positions.


PDB
Weekly update
Antibody chains identified using abnum
Heavy and Light chains paired
Antigens identified and paired with Fab
Experimental details
CDR clustering
IMGT gene annotations
Chothia numbering
Affinity data
VH-VL orientation
SAbDab
1600+ Antibody PDB structures
190 with curated affinity value
~3000 Fv structures
~ 6 new structures weekly
Easy dataset creation
Single structure analysis
VL
VH
Searches
PDB code
Experimental/ Structural/Sequence Property (Advanced Search)
Non-redundant structure search (antibody & antigen)
CDR cluster
CDR length or Sequence
VH-VL orientation (ABangle)
Template/Sequence search
List of selected structures. Bulk or individual download of dataset including:
PDB structure
Re-numbered structure
Structural annotations e.g. - H-L chain pairings - Bound-state - CDR clusters
Sequence / gene annotations e.g. - Ab/Ag Species - Chain type - IMGT subgroup
Experimental annotations. e.g. - Resolution - Method - Affinity ($K_d$)

Antibody Depositions in the PDB

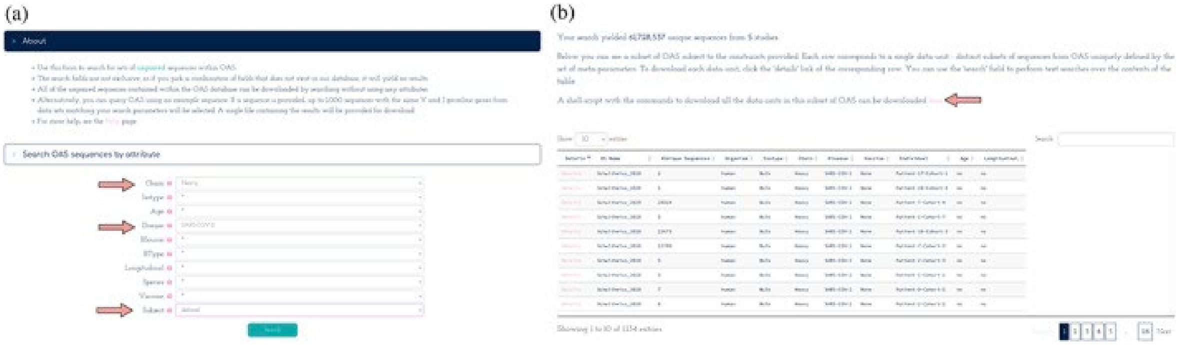


(SAbDab. accessed Feb. 2022)

**Figure 3.2:** SAbDab's workflow. New structures from the PDB are weekly analyzed to find antibody chains. These structures are then annotated with a number of properties and stored in SAbDab. Users may access and select this data using a number of different criteria. [James Dunbar et al, 2013]

## Observed Antibody Space (OAS)

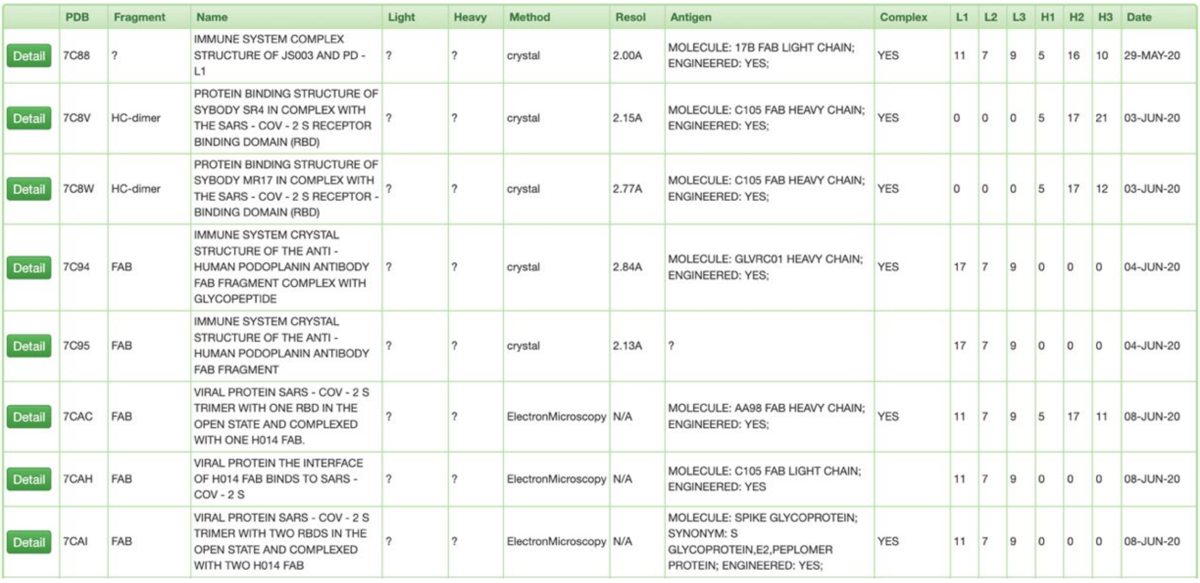
The Observed Antibody Space (OAS) database was created in 2018 to offer clean, annotated, and translated repertoire data. Driven by increasing volume of data and the appearance of paired (VH/VL) sequence data during last 4 years, OAS became accessible via a web-server [1], with standardized search parameters and sequence-based search option, to provide 1.5 billion unpaired sequences from 80 studies, including recent studies featuring SARS-CoV-2 data, and 172, 723 paired sequencing data from five studies. Providing the nucleotides for the VH/VL chains, the database also contains additional sequence annotations, such as the antibodies junction sequence and whether it is a productive sequence during wet-lab experiments, allowing for a fast initial query of 1,000 antibody sequences similar to a given sequence of interest.



Chain: Heavy
Isotype: *
Age: *
Disease: SARS-COV-2
BSource: *
BType: *
Longitudinal: *
Species: *
Vaccine: *
Subject: default

**Figure 3.3:** Downloading from OAS. (a) The sequence search tab for unpaired sequences, with the search options filled for heavy chain sequences from SARS-CoV-2 infected patients (shown with red arrows). (b) The search result, with each data unit matching the search and a downloadable link containing the links for the relevant data units (with a red arrow). [Olsen et al, 2022, OAS]

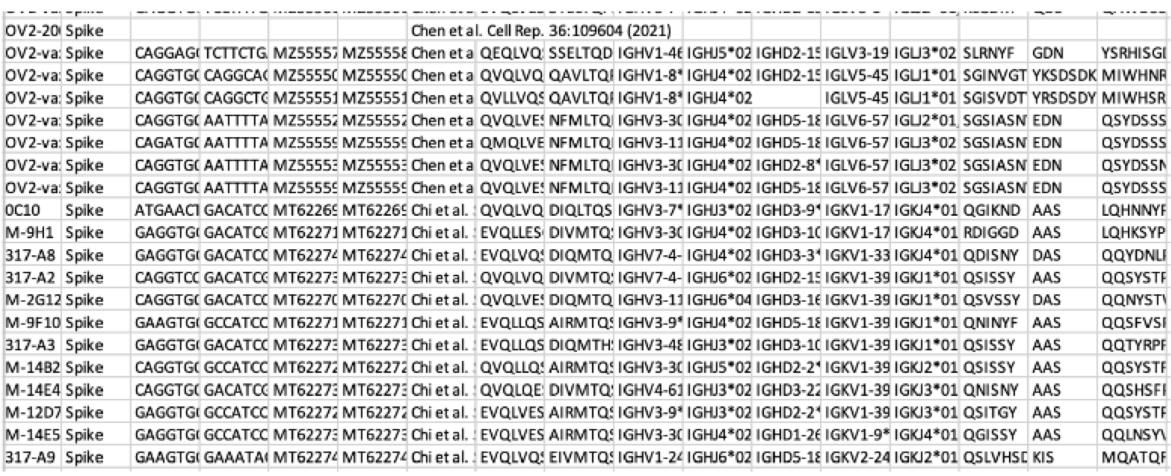
## Antigen-specific repertoire



| | PDB | Fragment | Name | Light | Heavy | Method | Resol | Antigen | Complex | L1 | L2 | L3 | H1 | H2 | H3 | Date |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Detail | 7C88 | ? | IMMUNE SYSTEM COMPLEX STRUCTURE OF JS003 AND PD - L1 | ? | ? | crystal | 2.00A | MOLECULE: 17B FAB LIGHT CHAIN; ENGINEERED: YES; | YES | 11 | 7 | 9 | 5 | 16 | 10 | 29-MAY-20 |
| Detail | 7C8V | HC-dimer | PROTEIN BINDING STRUCTURE OF SYBODY SR4 IN COMPLEX WITH THE SARS - COV - 2 S RECEPTOR BINDING DOMAIN (RBD) | ? | ? | crystal | 2.15A | MOLECULE: C105 FAB HEAVY CHAIN; ENGINEERED: YES; | YES | 0 | 0 | 0 | 5 | 17 | 21 | 03-JUN-20 |
| Detail | 7C8W | HC-dimer | PROTEIN BINDING STRUCTURE OF SYBODY MR17 IN COMPLEX WITH THE SARS - COV - 2 S RECEPTOR - BINDING DOMAIN (RBD) | ? | ? | crystal | 2.77A | MOLECULE: C105 FAB HEAVY CHAIN; ENGINEERED: YES; | YES | 0 | 0 | 0 | 5 | 17 | 12 | 03-JUN-20 |
| Detail | 7C94 | FAB | IMMUNE SYSTEM CRYSTAL STRUCTURE OF THE ANTI - HUMAN PODOPLANIN ANTIBODY FAB FRAGMENT COMPLEX WITH GLYCOPEPTIDE | ? | ? | crystal | 2.84A | MOLECULE: GLVRC01 HEAVY CHAIN; ENGINEERED: YES; | YES | 17 | 7 | 9 | 0 | 0 | 0 | 04-JUN-20 |
| Detail | 7C95 | FAB | IMMUNE SYSTEM CRYSTAL STRUCTURE OF THE ANTI - HUMAN PODOPLANIN ANTIBODY FAB FRAGMENT | ? | ? | crystal | 2.13A | ? | | 17 | 7 | 9 | 0 | 0 | 0 | 04-JUN-20 |
| Detail | 7CAC | FAB | VIRAL PROTEIN SARS - COV - 2 S TRIMER WITH ONE RBD IN THE OPEN STATE AND COMPLEXED WITH ONE H014 FAB. | ? | ? | ElectronMicroscopy | N/A | MOLECULE: AA98 FAB HEAVY CHAIN; ENGINEERED: YES; | YES | 11 | 7 | 9 | 5 | 17 | 11 | 08-JUN-20 |
| Detail | 7CAH | FAB | VIRAL PROTEIN THE INTERFACE OF H014 FAB BINDS TO SARS - COV - 2 S | ? | ? | ElectronMicroscopy | N/A | MOLECULE: C105 FAB LIGHT CHAIN; ENGINEERED: YES | | 11 | 7 | 9 | 0 | 0 | 0 | 08-JUN-20 |
| Detail | 7CAI | FAB | VIRAL PROTEIN SARS - COV - 2 S TRIMER WITH TWO RBDS IN THE OPEN STATE AND COMPLEXED WITH TWO H014 FAB | ? | ? | ElectronMicroscopy | N/A | MOLECULE: SPIKE GLYCOPROTEIN; SYNONYM: S GLYCOPROTEIN,E2,PEPLOMER PROTEIN; ENGINEERED: YES; | YES | 11 | 7 | 9 | 0 | 0 | 0 | 08-JUN-20 |

**Figure 3.4:** Summary of Antibody Crystal Structures extracted from Protein Data Bank (PBD) which are based on Observed Antibody Space (OAS) sequences

Since many sequence features of public antibody responses to different foreign viruses can be observed in Observed Antibody Space (OAS) [1] and Structural Antibody Database (SAbDab) [2], we postulate that the dataset is sufficiently large for gathering available antigen-specific antibodies for training the model. The preprocessing stage includes 172,723 filtered paired sequences with appropriate target diseases and organisms, along with 6,118 antibody structures available in the SAbDab. Since it is important to identify different antigens for distinguishing antibodies during the model training, we aim to retrieve 6,273 unique SARS-CoV-2 antibodies from OAS [1], 5,547 unique Human Immunodeficiency Virus (HIV) antibodies, and 2,204 unique influenza hemagglutinin (HA) antibodies from GenBank [22], with complete information for all six CDR sequences and germline expressions. Among different antigens, those were mainly chosen because of large number of published sequences and expressible antibodies binding to them.



| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OV2-20 | Spike | | | | | Chen et al. Cell Rep. 36:109604 (2021) | | | | | | | | | | |
| OV2-va | Spike | CAGGAG | TCTTCTG | MZ5555 | MZ5555 | Chen et a | QEQLVQ | SSELTQD | IGHV1-4 | IGHJ5*02 | IGHD2-1 | IGLV3-19 | IGLJ3*02 | SLRNYF | GDN | YSRHISG |
| OV2-va | Spike | CAGGTG | CAGGCA | MZ5555 | MZ5555 | Chen et a | QVQLVQ | QAVLTQ | IGHV1-8 | IGHJ4*02 | IGHD2-1 | IGLV5-45 | IGLJ1*01 | SGINVGT | YKSDSDK | MIWHN |
| OV2-va | Spike | CAGGTG | CAGGCT | MZ55551 | MZ55551 | Chen et a | QVLLVQ | QAVLTQ | IGHV1-8 | IGHJ4*02 | | IGLV5-45 | IGLJ1*01 | SGISVDT | YRSDSDY | MIWHSR |
| OV2-va | Spike | CAGGTG | AATTTTA | MZ5555 | MZ5555 | Chen et a | QVQLVE | NFMLTQ | IGHV3-3 | IGHJ4*02 | IGHD5-1 | IGLV6-57 | IGLJ2*01 | SGSIASN | EDN | QSYDSSS |
| OV2-va | Spike | CAGATG | AATTTTA | MZ5555 | MZ5555 | Chen et a | QMQLVE | NFMLTQ | IGHV3-11 | IGHJ4*02 | IGHD5-1 | IGLV6-57 | IGLJ3*02 | SGSIASN | EDN | QSYDSSS |
| OV2-va | Spike | CAGGTG | AATTTTA | MZ5555 | MZ5555 | Chen et a | QVQLVE | NFMLTQ | IGHV3-3 | IGHJ4*02 | IGHD2-8 | IGLV6-57 | IGLJ3*02 | SGSIASN | EDN | QSYDSSN |
| OV2-va | Spike | CAGGTG | AATTTTA | MZ5555 | MZ5555 | Chen et a | QVQLVE | NFMLTQ | IGHV3-11 | IGHJ4*02 | IGHD5-1 | IGLV6-57 | IGLJ3*02 | SGSIASN | EDN | QSYDSSS |
| 0C10 | Spike | ATGAAC | GACATC | MT6226 | MT6226 | Chi et al. | QVQLVQ | DIQLTQS | IGHV3-7 | IGHJ3*02 | IGHD3-9 | IGKV1-17 | IGKJ4*01 | QGIKND | AAS | LQHNNY |
| M-9H1 | Spike | GAGGTG | GACATC | MT62271 | MT62271 | Chi et al. | EVQLLES | DIVMTQ | IGHV3-3 | IGHJ4*02 | IGHD3-1 | IGKV1-17 | IGKJ4*01 | RDIGGD | AAS | LQHKSYP |
| 317-A8 | Spike | GAGGTG | GACATC | MT6227 | MT6227 | Chi et al. | EVQLVQ | DIQMTQ | IGHV7-4- | IGHJ4*02 | IGHD3-3 | IGKV1-33 | IGKJ4*01 | QDISNY | DAS | QQYDNL |
| 317-A2 | Spike | CAGGTC | GACATC | MT6227 | MT6227 | Chi et al. | QVQLVQ | DIVMTQ | IGHV7-4- | IGHJ6*02 | IGHD2-1 | IGKV1-39 | IGKJ1*01 | QSISSY | AAS | QQSYST |
| M-2G12 | Spike | CAGGTG | GACATC | MT6227 | MT6227 | Chi et al. | QVQLVE | DIQMTQ | IGHV3-11 | IGHJ6*04 | IGHD3-1 | IGKV1-39 | IGKJ1*01 | QSVSSY | DAS | QQNYST |
| M-9F10 | Spike | GAAGTG | GCCATCC | MT62271 | MT62271 | Chi et al. | EVQLLQS | AIRMTQ | IGHV3-9 | IGHJ4*02 | IGHD5-1 | IGKV1-39 | IGKJ1*01 | QNINYF | AAS | QQSFVS |
| 317-A3 | Spike | GAGGTG | GACATC | MT6227 | MT6227 | Chi et al. | EVQLLQS | DIQMTH | IGHV3-4 | IGHJ3*02 | IGHD3-1 | IGKV1-39 | IGKJ1*01 | QSISSY | AAS | QQTYRP |
| M-14B2 | Spike | CAGGTG | GCCATCC | MT6227 | MT6227 | Chi et al. | QVQLLQ | AIRMTQ | IGHV3-3 | IGHJ5*02 | IGHD2-2 | IGKV1-39 | IGKJ2*01 | QSISSY | AAS | QQSYST |
| M-14E4 | Spike | CAGGTG | GACATC | MT6227 | MT6227 | Chi et al. | QVQLQE | DIVMTQ | IGHV4-61 | IGHJ3*02 | IGHD3-2 | IGKV1-39 | IGKJ3*01 | QNISNY | AAS | QQSHSF |
| M-12D7 | Spike | GAGGTG | GCCATCC | MT6227 | MT6227 | Chi et al. | EVQLVES | AIRMTQ | IGHV3-9 | IGHJ3*02 | IGHD2-2 | IGKV1-39 | IGKJ3*01 | QSITGY | AAS | QQSYST |
| M-14E5 | Spike | GAGGTG | GCCATCC | MT6227 | MT6227 | Chi et al. | EVQLVES | AIRMTQ | IGHV3-3 | IGHJ4*02 | IGHD1-2 | IGKV1-9* | IGKJ4*01 | QGISSY | AAS | QQLNSY |
| 317-A9 | Spike | GAAGTG | GAAATA | MT6227 | MT6227 | Chi et al. | EVQLVQ | EIVMTQ | IGHV1-2 | IGHJ6*02 | IGHD5-1 | IGKV2-24 | IGKJ2*01 | QSLVHS | KIS | MQATQ |

**Figure 3.5:** A dataset consisting of 6,273 SARS-CoV-2 targeting antibodies with full sequence & germline expressions

## Structural dataset augmentation

We seek to train the model on as many immunoglobulin structures as possible. From the Structural Antibody Database (SAbDab) [2], we obtain 6,285 structures consisting of paired antibodies and single-chain nanobodies. Given the remarkable success of AlphaFold for modeling both protein monomers and complexes [21], we additionally explore the use of data augmentation to produce structures for training.

To produce a diverse set of structures for data augmentation, we clustered the paired and unpaired partitions of the Observed Antibody Space [1] at 40 % and 70 % sequence identity, respectively. This clustering results in 16,100 paired sequences and 26,900 unpaired sequences. We predict structures for both sets of sequences using the original AlphaFold model [21].

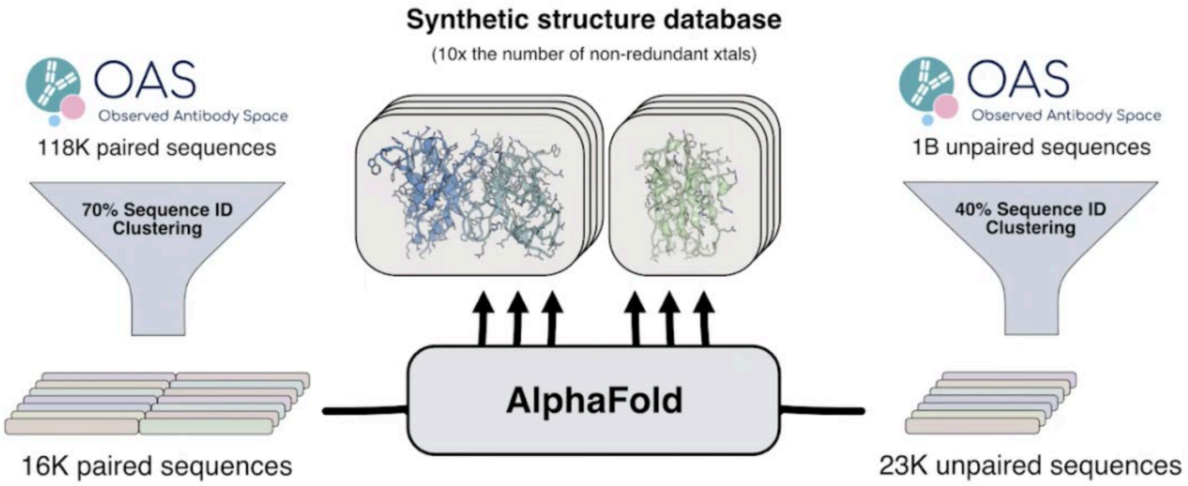


**Figure 3.6:** AlphaFold is used to create a synthetic structure dataset from natural antibody sequences

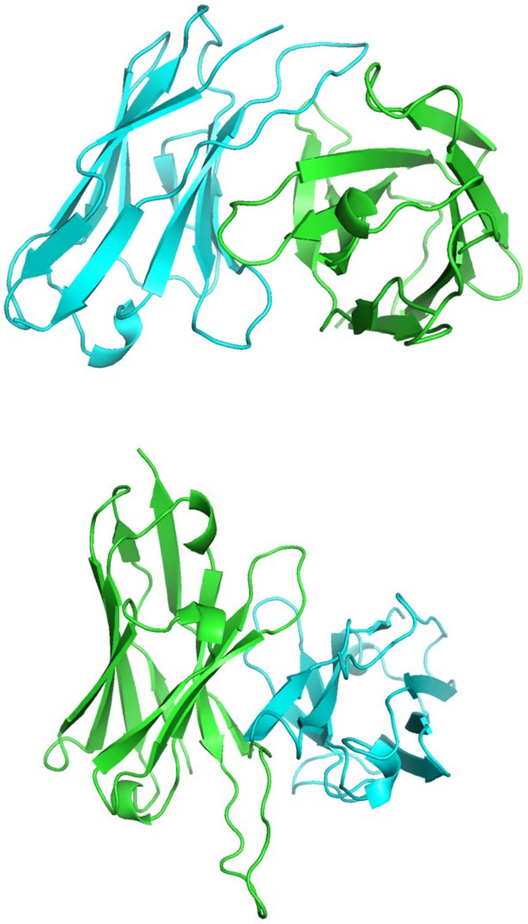


**Figure 3.7:** AlphaFold-predicted antibody structures for SARS-CoV-2 Omicron (upper) & HIV (bottom)

## 4. Approach & Methods

After cleaning up redundant sequences, clustering down to more manageable population of immunoglobulins, and gathering synthetic crystal structures from antibody prediction networks [7, 8, 21], we explore a structure-based pretraining model for antibodies which efficiently incorporates both amino acid representations and structural information. Furthermore, we are going to utilize a contrastive learning framework with augmentation functions to discover substructures in different antibodies, serving as a crucial step for allowing self-supervised learning on antibody structures.

### Geometry-Aware Relational Graph Neural Network

Considering antibody structures, the following model aims to learn encoded characteristics of spatial and chemical information. These representations must be invariant under 3D space of translations and rotations. To achieve this major requirement, we will first construct our antibody graph based on spatial features invariant under these transformations.

### Antibody graph construction

We represent the structure of an antibody as a residue-level relational graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$, in which $\mathcal{V}$ and $\mathcal{E}$ denote the set of nodes and edges respectively, and $\mathcal{R}$ is the set of edge types. We use $(i, j, r)$ to denote the edge from node $i$ to node $j$ with type $r$, and $n$, $m$ denote the number of nodes and edges, respectively. In this project, every node of the antibody graph will indicate the residue's alpha carbon with the 3D coordinates of all nodes $x \in \mathbb{R}^{n\times 3}$. We are going to utilize $\boldsymbol{f}_i$ and $\boldsymbol{f}_{(i,j,r)}$ to represent the feature for node $i$ and edge $(i, j, r)$, respectively.

Subsequently, 3 different types of directed edges will be added to these graphs: sequential, radius, and K-nearest neighbor edges. Among them, sequential edges will be further considered into 5 edge kinds dependent on the relative sequential distance $d \in \{-2, -1, 0, 1, 2\}$ between two end nodes, in which we will connect sequential edges only between the nodes within the sequential distance $d = 2$. Those edge types will reflect distinctive geometric characteristics, where all combined result in a comprehensive featurization of antibodies.

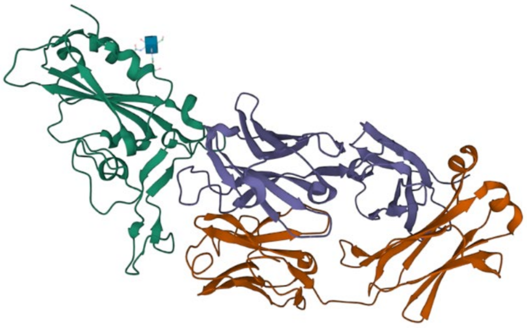


**Figure 4.1:** 7BWJ – crystal structure of SARS-COV-2 antibody from Protein Data Bank

### Relational graph convolutional layer

Based on the protein graph construction from the previous section, we will use a Graph Neural Network (GNN) to obtain per-residue and whole-protein representations. One baseline example of GNNs is the Graph Convolution Networks (GCN) [23], where messages are computed by multiplying node features with a convolutional kernel matrix shared among all edges. In order to increase the model capacity in protein modeling, Intrinsic-Extrinsic Convolution (IEConv) [24] introduced a learnable kernel function on edge features. With this approach, $m$ different kernel matrices would be applied on different edges, which results in a good performance but induces high memory costs.

To balance model capacity and memory cost, we will utilize a relational graph convolutional neural network [25] to learn graph representations, in which a convolutional kernel matrix is shared within each edge type and there are $|\mathcal{R}|$ different kernel matrices overall. Mathematically, the relational graph convolutional layer used in this work will be defined as

$$\boldsymbol{h}_i^{(0)} = \boldsymbol{f}_i,\ u_i^{(l)} = \sigma\left(\mathrm{BN}\left(\sum_{r\in\mathcal{R}} W_r \sum_{j\in\mathcal{N}_r(i)} h_j^{(l-1)}\right)\right),\ \boldsymbol{h}_i^{(l)} = h_i^{(l-1)} + u_i^{(l)}$$

More specifically, we will use node features $f_i$ as initial representations. Then, taking into account the node representation of $h_i^{(l)}$ for node $i$ at the $l$-th layer, we will compute updated node representation $u_i^{(l)}$ by aggregating neighboring nodes' features from $\mathcal{N}_r(i)$, in which

$\mathcal{N}_r(i) = \{j \in \mathcal{V} \mid (j, i, r) \in \mathcal{E}\}$ indicates the neighborhood of node $i$ with the edge type $r$, and $\boldsymbol{W}_r$ represents the learnable convolutional kernel matrix for edge type $r$. For this relational graph construction, BN stands for a batch normalization layer and we will use a ReLU function as the activation $\sigma(\cdot)$. At the end, we will update $h_i^{(l)}$ with $u_i^{(l)}$ and add a residual connection from the last layer.

## Edge Message Passing Layer

Reviewing the literature of molecular representation learning, we can observe the major importance of geometric encoders to explicitly modeling interactions between edges. For instance, Directional Message Passing Neural Network (DimeNet) [26] considers a 2D-spherical Fourier-Bessel basis function to indicate angles between 2 edges and pass messages between edges. AlphaFold2 [27] leverages the triangle attention designed for transformers to model pair representations. Encouraged from this literature, in this project, we will use a variant of Geometry-Aware Relational Graph Neural Network which is enhanced with an edge message passing layer. The edge message passing layer can be considered as a sparse version of the pair representation update designated for GNNs (Graph Neural Networks). The main goal is to model the dependency between different interactions of a residue with other sequentially/spatially adjacent residues.

Mathematically, we will first build a relational graph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}', \mathcal{R}')$ among edges, mostly inspired from the old literature [28]. Every node in the graph $\mathcal{G}'$ corresponds to the edge of the original graph $\mathcal{G}$. $\mathcal{G}'$ links edge $(i, j, r_1)$ in the original graph to edge $(w, k, r_2)$ if and only if $j = w$ and $i \neq k$. The specific kind of this edge is obtained by the angle between $(i, j, r_1)$ and $(w, k, r_2)$. To make computations easier, we are going to divide the range $[0, \pi]$ into eight bins and utilize the bin's index as the edge type.

Then, we will apply a similar relational graph convolutional network on the graph $\mathcal{G}'$ to determine the message function for every edge. Specifically, the edge message passing layer for this model will be defined as follows:

$$\boldsymbol{m}_{(i,j,r_1)}^{(0)} = \boldsymbol{f}_{(i,j,r_1)}, \ \boldsymbol{m}_{(i,j,r_1)}^{(l)} = \sigma\left( \mathrm{BN}\left( \sum_{r \in \mathcal{R}'} \boldsymbol{W}'_r \sum_{(w,k,r_2) \in \mathcal{N}'_r((i,j,r_1))} \boldsymbol{m}_{(w,k,r_2)}^{(l-1)} \right) \right)$$

In this case, we denote $m_{(i,j,r_1)}^{(l)}$ to be the message function for edge $(i,j,r_1)$ in the $l$-th layer. Similar to the previous graph convolution equation, the message function for edge $(i,j,r_1)$ will be updated by combining neighbor features $\mathcal{N}_r'((i,j,r_1))$, in which $\mathcal{N}_r'((i,j,r_1)) = \{(w,k,r_2) \in \mathcal{V}' \mid ((w,k,r_2),(i,j,r_1),r) \in \mathcal{E}'\}$ indicates the set of incoming edges of $(i,j,r_1)$ with relation type $r$ in the graph $\mathcal{G}'$.

At the end, we will replace the aggregation function from graph convolution equation in the original graph with the following (referenced to previous chapter):

$$u_i^{(l)} = \sigma\left(\mathrm{BN}\left(\sum_{r\in\mathcal{R}} W_r \sum_{j\in\mathcal{N}_r(i)} \left(h_j^{(l-1)} + \mathrm{FC}\left(\boldsymbol{m}_{(j,i,r)}^{(l)}\right)\right)\right)\right)$$

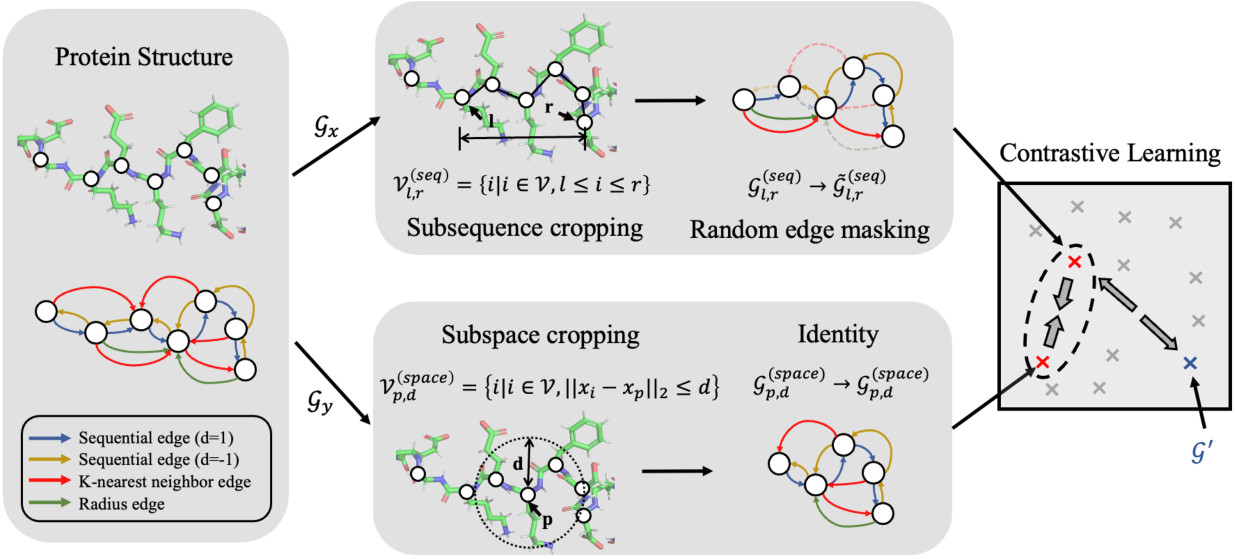
where FC$(\cdot)$ indicates a linear transformation upon the message function.

## 5. Main results

In this section, we will first explore how to boost antibody representation learning via self-supervised pretraining on a huge number of unlabeled antibody structures, and then present the antigen-specificity results from those representations.

### Self-supervised pretraining methods

Even though self-supervised pretraining has proven to be effective in multiple fields, application of those methods to antibody representation learning is not easy because of the difficulty of incorporating both biochemical and spatial characteristics in antibody structures. In order to approach this challenge, we use a multi-view contrastive learning method with augmentation functions to discover correlated co-occurrence of antibody sub-structures and align their representations in the latent space.



Protein Structure
Sequential edge (d=1)
Sequential edge (d=-1)
K-nearest neighbor edge
Radius edge
$\mathcal{G}_x$
$\mathcal{G}_y$
l
r
$\mathcal{V}^{(seq)}_{l,r} = \{i | i \in \mathcal{V}, l \leq i \leq r\}$
Subsequence cropping
$\mathcal{G}^{(seq)}_{l,r} \rightarrow \tilde{\mathcal{G}}^{(seq)}_{l,r}$
Random edge masking
Subspace cropping
Identity
$\mathcal{V}^{(space)}_{p,d} = \{i | i \in \mathcal{V}, ||x_i - x_p||_2 \leq d\}$
$\mathcal{G}^{(space)}_{p,d} \rightarrow \mathcal{G}^{(space)}_{p,d}$
d
p
Contrastive Learning
$\mathcal{G}'$

**Figure 5.1**: Multiview Contrastive Learning: For each antibody, we first construct the residue graph $\mathcal{G}$ based on the structural information. Then, two sub-views $\mathcal{G}_x$ and $\mathcal{G}_y$ of the antibody are generated by randomly choosing the sampling scheme and noise function. For $\mathcal{G}_x$, we extract a subsequence and then perform random edge masking where dash lines represent masked edges. For $\mathcal{G}_y$, we apply a subspace cropping and then keep the subspace graph $\mathcal{G}^{(\text{space})}_{p,d}$ intact. At the end, a contrastive learning loss is optimized to maximize the similarity between $\mathcal{G}_x$ and $\mathcal{G}_y$ in the latent space while minimizing its similarity with a negative sample $\mathcal{G}'$

### Multiview contrastive learning

Inspired by recent contrastive learning methods [29, 30] and the evolutional history of antibody substructures within the same folded motif [31], we will explore a framework that aims to preserve the similarity between those correlated substructures before and after mapping to a low-dimensional latent space. Particularly, considering a similarity measurement that is applied in the latent space, biologically-correlated antibody substructures will be embedded close to each other, whereas the unrelated ones will be mapped far apart from each other. Figure 5.1 represents the high-level idea of this approach.

### Construction of antibody substructures

Considering an antibody graph $\mathcal{G}$, we will apply 2 distinctive sampling schemes for reflecting substructure views. The first one is subsequence cropping, which randomly samples a left residue $l$ and a right residue $r$ and aggregates all amino acid residues from $l$ to $r$. This sampling scheme aims to capture antibody domains, consecutive subsequences that reappear in different antibodies along with their functionalities [32]. But, just sampling antibody subsequences will not fully capture the 3D structural representation from antibody data. Thus, we further apply a subspace cropping scheme that explores spatially structurally correlated motifs in antibodies. We will randomly sample an amino acid residue $p$ as the center and consider all residues within a Euclidean ball with a predefined radius $d$. For the two-sampling schematic, we will take the corresponding subgraphs from the antibody residue graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$. Particularly, the subsequence graph $\mathcal{G}_{l,r}^{(\text{seq})}$ and the subspace graph $\mathcal{G}_{p,d}^{(\text{space})}$ are defined as follows:

$$\mathcal{V}_{l,r}^{(\text{seq})} = \{i \mid i \in \mathcal{V}, l \leq i \leq r\},\ \mathcal{E}_{l,r}^{(\text{seq})} = \{(i,j,r) \mid (i,j,r) \in \mathcal{E}, i \in \mathcal{V}_{l,r}^{(\text{seq})}, j \in \mathcal{V}_{l,r}^{(\text{seq})}\},$$

$$\mathcal{V}_{p,d}^{(\text{space})} = \left\{i \mid i \in \mathcal{V}, \|\boldsymbol{x}_i - \boldsymbol{x}_p\|_2 \leq d\right\},\ \mathcal{E}_{p,d}^{(\text{space})} = \{(i,j,r) \mid (i,j,r) \in \mathcal{E}, i \in \mathcal{V}_{p,d}^{(\text{space})}, j \in \mathcal{V}_{p,d}^{(\text{space})}\},$$

$$\text{where } \mathcal{G}_{l,r}^{(\text{seq})} = \left(\mathcal{V}_{l,r}^{(\text{seq})}, \mathcal{E}_{l,r}^{(\text{seq})}, \mathcal{R}\right) \text{ and } \mathcal{G}_{p,d}^{(\text{space})} = \left(\mathcal{V}_{p,d}^{(\text{space})}, \mathcal{E}_{p,d}^{(\text{space}))} \mathcal{R}\right)$$

Referencing the usual practice of self-supervised learning [29], once the substructures are sampled, we will apply a noise function to construct more diverse views which in turn, will improve the learned representations. In this project, we are going to use 2 noise functions: “Identity” which applies no transformation at all, and “Random edge masking” which randomly masks every edge with a probability $p = 0.20$

## Contrastive learning

We use a Simple Framework for Contrastive Learning of Visual Representations (SimCLR) [29] to optimize a contrastive loss function and in turn, maximize the mutual information between biologically correlated views. Specifically, we sample views $\mathcal{G}_x$ and $\mathcal{G}_y$ for every antibody $\mathcal{G}$, first by randomly choosing one sampling scheme for extracting substructures and then, randomly selecting one of the two noise functions (i.e. "Identity" or "Random edge masking") with equal probability. The graph representations $\boldsymbol{h}_x$ and $\boldsymbol{h}_y$ of two views will be obtained by applying the structure-based encoder we described in "Approach & Methods". Then, 2-layer MLP (Multi-Layer Perceptron) projection head is used to map the representations to a lower-dimensional space, indicated as $\boldsymbol{z}_x$ and $\boldsymbol{z}_y$. At the end, an InfoNCE (Noise-Contrastive Estimation) loss function is defined by differentiating views from the same or different antibodies using their similarities [33]. Specifically, for a positive pair $x$ and $y$, we treat views from other antibodies in the same mini-batch as negative pairs. Mathematically, the loss function for a positive pair of views $x$ and $y$ will be written as:

$$\mathcal{L}_{x,y} = -\log \frac{\exp\left(\mathrm{sim}\left(\boldsymbol{z}_x, \boldsymbol{z}_y\right)/\tau\right)}{\sum_{k=1}^{2B} \mathbb{1}_{[k \neq x]} \exp\left(\mathrm{sim}\left(\boldsymbol{z}_y, \boldsymbol{z}_k\right)/\tau\right)}$$

Here, $B$ denotes the batch size, $\tau$ indicates the temp1e2qserature, $\mathbb{1}_{[k \neq x]} \in \{0,1\}$ is an indicator function that is equal to 1 if and only if $k \neq x$. The similarity function $\mathrm{sim}(\boldsymbol{u}, \boldsymbol{v})$ is obtained by the cosine similarity between $\boldsymbol{u}$ and $\boldsymbol{v}$.

## Evaluation metrics

Now, we are going to discuss the evaluation metrics for antigen-specificity prediction. As a revision, our goal is to answer the following question: whether an antibody binds to SARS-CoV-2, HIV, or Hemagglutinin Influenza (HA), which can be regarded as a multi-class binary classification task.

The 1st metric, antibody-centric maximum F-score $F_{\max}$, is obtained by first computing the precision and recall for each antibody and then taking the average score over all antibodies. Mathematically, for a given target antibody $i$ and a decision threshold $t \in [0,1]$, the precision and recall will be evaluated as follows:

$$\text{precision}_i(t) = \frac{\sum_f \mathbb{1}[f \in P_i(t) \cap T_i]}{\sum_f \mathbb{1}[f \in P_i(t)]},$$

and

$$\text{recall}_i(t) = \frac{\sum_f \mathbb{1}[f \in P_i(t) \cap T_i]}{\sum_f \mathbb{1}[f \in T_i]},$$

in which $f$ is a function for the antigen-specificity, $T_i$ is a set of experimentally determined antigen terms for antibody $i, P_i(t)$ denotes the set of predicted terms for antibody $i$ with scores $\geq t$ and $\mathbb{1}[\cdot] \in \{0,1\}$ is an indicator function which is equal to 0 iff the condition is false.

Subsequently, the average precision and recall over all antibodies at threshold $t$ will be as:

$$\text{precision}(t) = \frac{1}{M(t)} \sum_i \text{precision}_i(t)$$

and

$$\text{recall}(t) = \frac{1}{N} \sum_i \text{recall}_i(t),$$

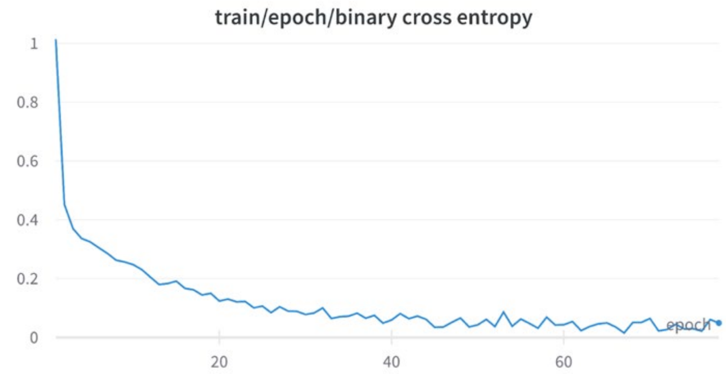
in which $N$ indicates the number of antibodies and $M(t)$ represents the number of antibodies on which at least 1 prediction was made above threshold $t$; in other words, $|P_i(t)| > 0$.

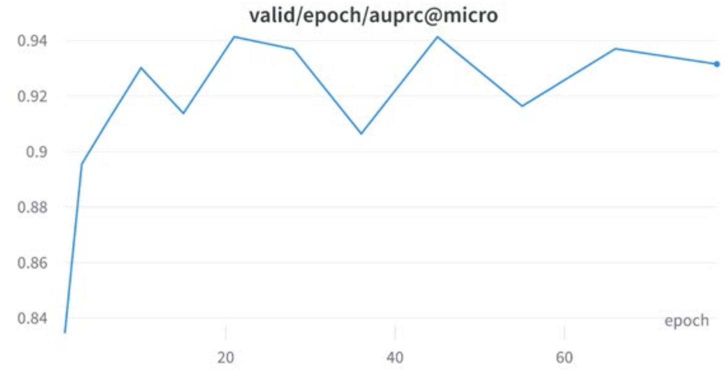
Aggregating those 2 evaluation measurements, the maximum F-score will be denoted as the maximum value of F-measure over all thresholds. In other words,

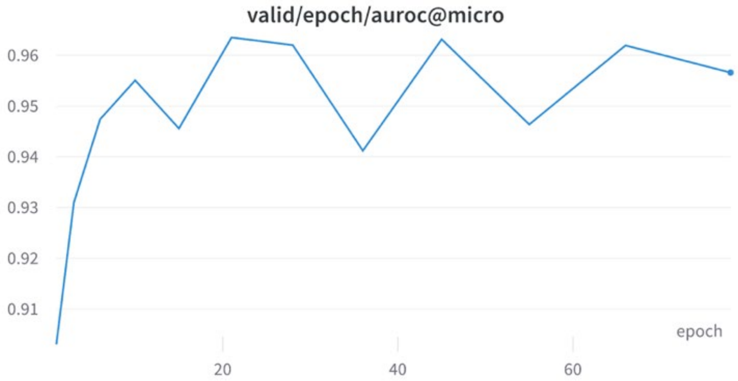
$$\text{F}_{\text{max}} = \max_t \left\{ \frac{2 \,*\, \text{precision}(t) \,*\, \text{recall}(t)}{\text{precision}(t) + \text{recall}(t)} \right\}$$

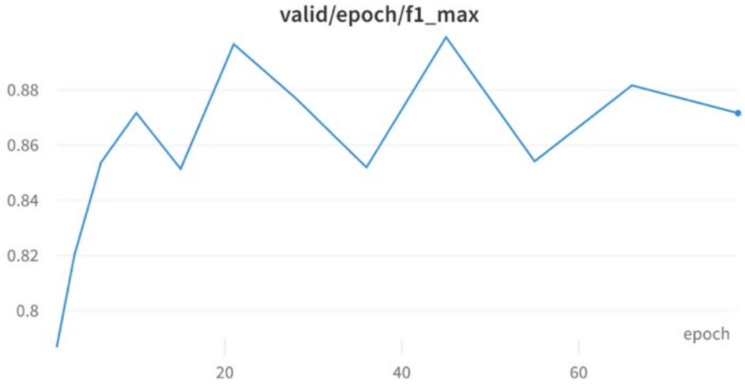
The 2nd metric, pair-centric Area Under Precision-Recall Curve $\text{AUPRC}_{\text{pair}}$ , is obtained as the average precision scores for all antibody-antigen pairs, which is actually the micro average precision score for multiple binary classification.

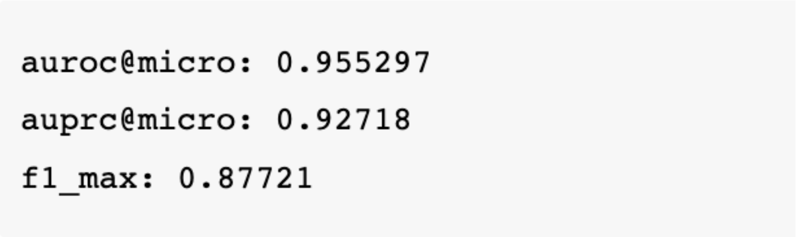
## Experimental results





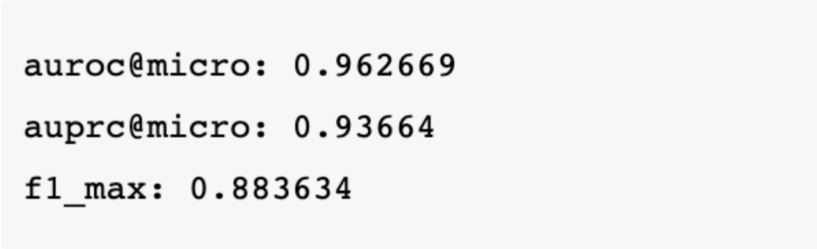






```
auroc@micro: 0.955297
auprc@micro: 0.92718
f1_max: 0.87721
```

**Geometry-Aware Relational Graph Neural Network**



```
auroc@micro: 0.962669
auprc@micro: 0.93664
f1_max: 0.883634
```

**Geometry-Aware Relational Graph Neural Network + Edge Message Passing Layer**

As seen from the above results, both of the explored methods indicate considerably good results for $F_{\max}$ measurement, in which binary cross entropy loss for training shows the desired decreasing curve throughout the 80 epochs. Moreover, Area Under Precision-Recall Curve (AUPRC), and Area Under Receiver Operating Characteristic (AUROC) reveal the fluctuating curves for the validation dataset, along with accurate evaluation metrics during their final epochs.

### Ablation studies

To analyze the contribution of different subcomponents in the explored methods, we will perform ablation studies on the antigen-specificity task. The results are shown in Table 5.1.

| Method | $\mathbf{F}_{\max}$ |
|---|---|
| Multiview contrastive learning | **0.884** |
| - subsequence + identity | 0.851 |
| - subspace+ identity | **0.862** |
| - subsequence + random edge masking | 0.858 |
| - subspace + random edge masking | **0.870** |

**Table 5.1:** Ablation studies on antibody-antigen datasets

# 6. Discussion

## Different augmentations in Multiview contrast

We explore the contribution of every augmentation approach introduced in the Multiview Contrast method (previous page). Instead of randomly sampling cropping and noise functions, we will pretrain our model with 4 deterministic combinations of augmentations, correspondingly. As indicated in Table 5.1, all the four combinations yield good results, which suggests arbitrary combinations of the introduced cropping and noise schemes can yield informative partial views of antibodies. Moreover, we can also see that the results of Subspace Cropping are usually better than those of Subsequence Cropping with different noise functions, concluding that it is an efficient method to utilize 3D information to extract meaningful antibody substructures.

## Protein latent space visualization

In order to evaluate the quality of the antibody embeddings learned by the pretraining method, we will visualize the latent space of the model pretrained by Multiview Contrast. Particularly, we will utilize the pretrained model to extract the embeddings of all the proteins from Alpha-Fold database, and these embeddings will be mapped to the 2-dimensional space by UMAP [34] for visualization. Referring to [35], we will highlight the 20 most common superfamilies within the database by distinct colors. The visualization results are indicated in Fig. 6.1. We can vividly observe that the pretrained model groups the same superfamily proteins together and divide the ones from different superfamilies apart. In fact, it succeeds in clearly separating three superfamilies, which are represented as Protein kinase superfamily, Cytochrome P450 family, and TRAFAC class myosin-kinesin ATPase superfamily.

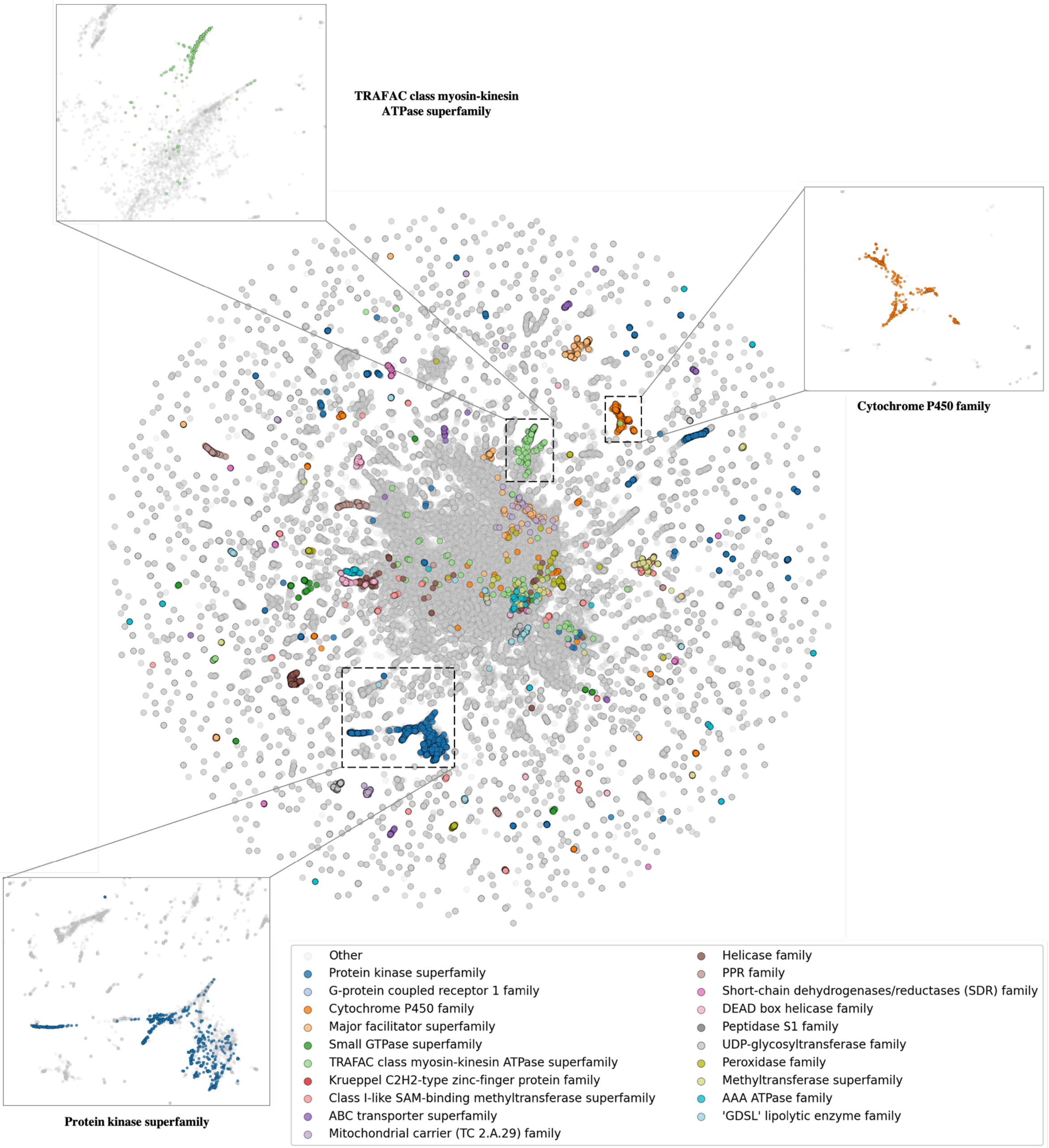


**Figure 6.1:** Latent space visualization of Multiview contrastive learning on the protein database

## 7. Future Research Plan & Proposal

Once we identify the epitope-paratope level interactions, binding sites, and neutralization activities available from antibody datasets, we can start experimenting with the benchmark architectures for contrastive representations based upon SimCSE [10] and SimCLR [11], along with appropriate graph-level embeddings for pathogenic antigens. Further data augmentation techniques for amino acid level representations may also be applied to improve the benchmark results of previous methods and efficiently create the massive antibody repertoire targeting new pathogens.

In recap, our summary for future work and ablation studies in this project can be summarized as follows:

- Obtain 3D surface/mesh-level representations of antibody structures using Geometric Deep Learning, including in silico synthetic datasets
- Propose a novel data augmentation mechanism for antibody data to produce additional synthetic structures for learning better representations using SimCSE and SimCLR algorithms.

## 8. Reference Literature

[1] Tobias H Olsen, Fergus Boyles, CharlotteMDeane, 2022, Observed Antibody Space: A diverse database of cleaned, annotated, and translated unpaired and paired antibody sequence

[2] James Dunbar et al, SAbDab: the structural antibody database, 2013

[3] Jared Adolf-Bryfogle, et al, 2018, RosettaAntibodyDesign (RAbD): A general framework for computational antibody design

[4] Christoffer Norn, Basile I. M. Wicky, David Juergens, Sergey Ovchinnikov, Protein sequence design by conformational landscape optimization, PNAS, 2021

[5] Yiquan Wang, Meng Yuan, Huibin Lv, Jian Peng, Ian A. Wilson, Nicholas C. Wu, A large-scale systematic survey reveals recurring molecular features of public antibody responses to SARS-CoV-2, Immunity, 2022

[6] Jeffrey A. Ruffolo, Jeffrey J. Gray, Jeremias Sulam, Deciphering antibody affinity maturation with language models and weakly supervised learning, MLSB, 2021

[7] RW Shuai, Jeffrey A. Ruffolo, Jeffrey J. Gray, Generative Language Modeling for Antibody Design, bioRxiv, 2021

[8] Jeffrey A. Ruffolo, Jeffrey J. Gray, Fast, accurate antibody structure prediction from deep learning on massive set of natural antibodies, Cell, 2022

[9] Wengong Jin, Jeremy Wohlwend, Regina Barzilay, Tommi Jaakkola, Iterative Refinement Graph Neural Network for Antibody Sequence-Structure Co-Design, arXiv, 2021

[10] Tianyu Gao, Xingcheng Yao, Danqi Chen, SimCSE: Simple Contrastive Learning of Sentence Embeddings, 2021

[11] Ting Chen, Simon Mohammad Norouzi, Geoffrey Hinton, A Simple Framework for Contrastive Learning of Visual Representations, 2020

[12] Jacob Devlin, Ming-Wei, Chang, Kenton Lee, Kristina Toutanova, BERT: Pretraining of Deep Bidirectional Transformers for Language Understanding, arXiv, 2018

[13] Ruei-Min Lu, Yu-Chyi Hwang, I-Ju Liu, Chi-Chiu Lee, Han-Zen Tsai, Hsin-Jung Li, Han-Chung Wu, Development of therapeutic antibodies for the treatment of diseases, PubMed Central, 2020

[14] Andrew R.M. Bradbury, Stefan DÅNubel, Achim Knappik, and Andreas Plückthund, Animal versus in vitro-derived antibodies: avoiding the extremes, PubMed Central, 2021

[15] Ethan C Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M Church, Unified rational protein engineering with sequence-based deep representation learning. Nature methods, 2019

[16] Jung-Eun Shin, Adam J Riesselman, Aaron W Kollasch, Conor McMahon, Elana Simon, Chris Sander, Aashish Manglik, Andrew C Kruse, and Debora S Marks, Protein design and variant prediction using autoregressive generative models, Nature communications, 2021

[17] John Ingraham, Vikas K Garg, Regina Barzilay, and Tommi Jaakkola, Generative models for graph-based protein design. Neural Information Processing Systems, 2019

[18] Alexey Strokach, David Becerra, Carles Corbi-Verge, Albert Perez-Riba, and Philip M Kim, Fast and flexible design of novel proteins using graph neural networks. BioRxiv, 2020

[19] Mostafa Karimi, Shaowen Zhu, Yue Cao, and Yang Shen. De novo protein design for novel folds using guided conditional wasserstein generative adversarial networks. Journal of Chemical Information and Modeling, 2020

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, Attention is all you need, Advances in Neural Information Processing Systems 30 (NIPS 2017)

[21] Jumper, J., Evans, R., Pritzel, A. et al. Highly accurate protein structure prediction with AlphaFold. Nature 596, 583–589 (2021)

[22] Benson DA, Cavanaugh M, Clark K, Karsch-Mizrachi I, Lipman DJ, Ostell J, et al. GenBank. Nucleic Acids Res. 2013;41(Database issue). PubMed PMID: 23193287; PubMed Central

[23] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In International Conference on Learning Representations, 2017.

[24] Vignesh Ram Somnath, Charlotte Bunne, and Andreas Krause. Multi-scale representation learning on proteins. Advances in Neural Information Processing Systems, 34, 2021.

[25] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In European semantic web conference, pp. 593–607. Springer, 2018.

[26] Johannes Klicpera, Janek Grob, and Stephan Günnemann. Directional message passing for molecular graphs. In International Conference on Learning Representations (ICLR), 2020.

[27] John Jumper, Richard Evans Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. Nature, 596(7873):583–589, 2021.

[28] Frank Harary and Robert Z Norman. Some properties of line digraphs. Rendiconti del circolo matematico di palermo, 9(2):161–168, 1960.

[29] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In International conference on machine learning, pp. 1597–1607. PMLR, 2020.

[30] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9729–9738, 2020.

[31] Craig O. Mackenzie and Gevorg Grigoryan. Protein structural motifs in prediction and design. Current opinion in structural biology, 44:161–167, 2017.

[32] Chris Paul Ponting and Robert R Russell. The natural history of protein domains. Annual review of biophysics and biomolecular structure, 31:45–71, 2002.

[33] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748, 2018.

[34] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426, 2018.

[35] Mehmet Akdel, Douglas EV Pires, Eduard Porta Pardo, Jürgen Jänes, Arthur O Zalevsky, Bálint Mészáros, Patrick Bryant, Lydia L Good, Roman A Laskowski, Gabriele Pozzati, et al. A structural biology community assessment of alphafold 2 applications. bioRxiv, 2021.