PREDICTING MORE INFECTIOUS VIRUS VARIANTS FOR PANDEMIC PREVENTION THROUGH DEEP LEARNING

Glenda Tan Hui En[1], KoayTze Erhn[1] and Shen Bingquan[2]

[1]Raffles Institution, Singapore
[2]DSO National Laboratories, Singapore

ABSTRACT

More infectious virus variants can arise from rapid mutations in their proteins, creating new infection waves. These variants can evade one's immune system and infect vaccinated individuals, lowering vaccine efficacy. Hence, to improve vaccine design, this project proposes Optimus PPIme – a deep learning approach to predict future, more infectious variants from an existing virus (exemplified by SARS-CoV-2). The approach comprises an algorithm which acts as a "virus" attacking a host cell. To increase infectivity, the "virus" mutates to bind better to the host's receptor. 2 algorithms were attempted – greedy search and beam search. The strength of this variant-host binding was then assessed by a transformer network we developed, with a high accuracy of 90%. With both components, beam search eventually proposed more infectious variants. Therefore, this approach can potentially enable researchers to develop vaccines that provide protection against future infectious variants before they emerge, pre-empting outbreaks and saving lives.

KEYWORDS

Virus Variants, Transformers, Deep Learning

1. BACKGROUND AND PURPOSE OF RESEARCH AREA

1.1. The Emergence of more Infectious Virus Variants

Background and Motivation: A small proportion of rapid mutations in viral genomes can significantly increase infectivity, leading to waves of new infections, which in turn create opportunities for more viral mutations to occur. This mechanism has prolonged the devastating pandemic caused by the novel severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). As variants of concern emerge, including Beta, Delta and Omicron [1], evidence points towards mutations in the SARS-CoV-2 spike glycoprotein that increase its binding affinity towards the human angiotensin-converting enzyme 2 (hACE2) receptor [2], thus raising transmissibility.

Challenges and Objectives: Currently, vaccines are the solution to reducing virus transmissibility by training one's immune system to develop a response against the virus. However, emergent virus variants can evade one's immune system and even infect vaccinated individuals, lowering vaccine efficacy. For instance, the Covid-19 Delta variant drastically lowered Pfizer BioNTech vaccine's efficacy from 93.7% to 39%, triggering global infection waves in 2021 [3, 4]. Unless vaccines are designed to combat both current and future more infectious variants, our pandemic battle will be a prolonged cat-and-mouse game where we struggle to develop booster shots to catch up with ever-mutating variants. Hence, we aim to use deep learning to predict future, more

DOI: 10.5121/ijaia.2022.13403

infectious virus variants. This can help researchers to prepare for vaccine production against these variants before they arise, pre-empting outbreaks and saving lives.

Contributions: In this paper, our contributions include:

a. Developing a deep learning approach, Optimus PPIme, that generates mutations from an existing virus protein (exemplified by SARS-CoV-2) to predict future, more infectious variants.
b. Developing a protein-protein interaction (PPI) transformer neural network, that can score the binding affinity between a virus protein and host receptor with a high test accuracy of 90%. Only protein primary sequences are needed as input.

In section 1, we introduce our Optimus PPIme approach and cite related work. Section 2 describes our research question and hypothesis while Section 3 documents the development of Optimus PPIme. Our results are shown in Section 4, while implications of our approach and future work are covered in Sections 5 and 6 respectively.

1.2. Our Deep Learning Approach – Optimus PPIme

Consider the following: A virus attacking a host cell aims to discover mutations that maximize its binding affinity to the host receptor, thereby increasing its infectivity. This is akin to a game character deciding on an optimal strategy to maximize its long-term reward –any action made at one time-step affects subsequent rewards. In the first context, our agent (the virus) can substitute any amino acid (AA) in its sequence of length L (L = 1273 for SARS-CoV-2) with 1 of the 20 AAs, giving rise to an action space of 20L. We exclude insertion and deletion mutations as these are less common in nature [5]. The environment (PPI transformer network) then outputs a PPI score for the proposed mutated protein (new state) and host receptor. The reward received is the change in PPI score (final – initial score of original virus protein S_0 and host receptor).

1.3. Related Work

PPIs are usually determined via tedious and costly high-throughput experimental methods, such as isothermal titration calorimetry and nuclear-magnetic resonance [6]. This necessitates the use of computational PPI models. However, many require 3D protein structures, which are harder to obtain than primary sequences. Even methods such as MuPIPR [7] –that only require primary sequences as inputs– fail to generalize to novel proteins. To address these, we propose a PPI transformer network that uses only primary sequences and generalizes to novel proteins.

Primary sequences can be represented as strings of letters denoting AAs. Protein motifs and domains (conserved functional patterns) [8] are also analogous to words and phrases. Furthermore, information is contained in primary sequences and natural sentences [9]. Such similarities make natural language processing (NLP) ideal for extracting protein features.

NLP tasks have seen state-of-the-art performance with the rise of transformers. These encoder-decoder networks adopt the self-attention mechanism, which relates different positions of a sequence to compute a rich representation of features [10]:

$$attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_K}}\right)V$$

Where Q, K, and V are the query, key and value matrices while d_K is the dimension of K. Given that interactions between different AAs in a protein sequence give rise to the protein's structure and properties, transformer encoders are suitable protein feature extractor networks.

Lastly, we define a similarity measure between S_0 and a generated variant protein with sequence alignment scores derived from Block Substitution Matrix 62 (BLOSUM62) [11]. The BLOSUM distance between 2 sequences, $S = s_1 \ldots s_L$ and $S' = s'_1 \ldots s'_L$, is given by [12]:

$$D(S, S') = \sum_{i=1}^{L} (B_{s_i s_i} - B_{s_i s'_i})$$

2. Hypothesis of Research

Our research question is: Is a PPI predictor environment sufficient for virus agents to predict future, more infectious variants?

We hypothesize that given an environment that scores the binding affinity for a virus-host PPI with high accuracy, agents can predict future, more infectious variants.

3. Methodology

3.1. Dataset Collection

We train our PPI transformer on 31,869 experimentally-validated positive virus-human (V-H) interactions from BioGRID [13] and VirHostNet [14], excluding PPIs with SARS-CoV-2. We reference 34,836 negative V-H interactions from [15], giving rise to 66,705 training examples.

For the 50 novel virus protein test PPIs, we fix hACE2 as the second input protein. Positive PPIs include the original SARS-CoV-2 spike and its 23 variants listed in CoVariants [16], while 26 negative PPIs were sampled from unseen non-coronavirus viral proteins in [14]. All sequences were extracted from UniProt [17] and tokenized with Keras TextVectorization. We added start, end of sentence (SOS, EOS) and class (CLS) tokens, padding up to 1,300 tokens.

3.2. The Environment – PPI Transformer Network

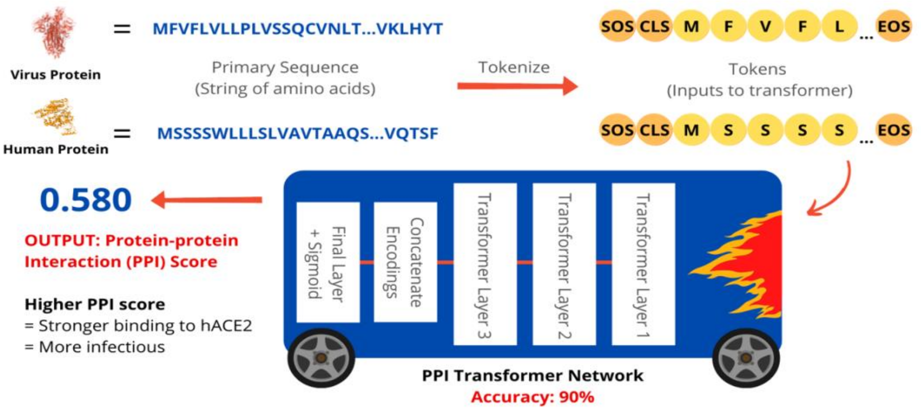


Figure 1. Overview of the PPI transformer network.

The inputs to our PPI transformer are the primary sequences of a virus and human receptor protein (represented by strings of amino acids). These strings of amino acids are then tokenized before they are fed to the transformer network. The transformer network then outputs a PPI score measuring how well the virus protein binds to the human receptor protein. The higher the PPI score, the stronger the virus-host binding and the more infectious the variant.

3.2.1. Experiment 1: Masked Language Modeling (MLM) Pre-training vs No MLM Pre-Training

Initially, our transformer predicted the same value of 0.99 for all test data (see Appendix), indicating its inability to learn. This may be due to unbalanced dependencies and amplified output perturbations caused by layer normalization and aggravated by the Adam optimizer. Since learning semantics of input sequences improves downstream task performance (transfer learning) [18, 19], we pre-train our transformer on Masked Language Modeling (MLM) and fine-tune on PPI. In each sequence, we mask a random 15% of tokens with a mask token 90% of the time, or a random token otherwise. MLM attempts to reconstruct the original sequence.

3.2.2. Experiment 2: Sharpness-Aware Minimization (SAM) vs No SAM

Today’s neural networks optimize on training loss, but this is inadequate in generalizing to unseen data (novel proteins for our PPI task) due to complex and non-convex training loss landscapes. SAM overcomes this by simultaneously minimizing loss value and loss sharpness. This algorithm enables a model to learn parameters whose entire neighbourhoods have uniformly low training loss values (low losses and curvature) rather than those with solely low loss values (low losses but complex curvature) [20, 21]. Hence, we determine the effects of SAM on generalizability. Using the model pre-trained with MLM in Experiment 1 as our baseline, 3 new models were trained with the addition of SAM on MLM pre-training only, PPI training only, and for both tasks (see Appendix for our implementation).

3.2.3. Experiment 3: Data Augmentation vs No Data Augmentation

Given that image augmentation improves image classifiers’ robustness [22], we aim to determine if augmenting protein sequences could similarly boost our PPI test accuracies. The models were trained with 3 different augmentation techniques [23] during MLM pre-training: substituting a random AA with alanine –a generally unreactive AA (Alanine Sub)– or an AA of the highest BLOSUM62 similarity score –the most likely AA that can replace the original AA (Dict Sub), and reversing the whole protein sequence (Reverse). 25% of proteins were augmented and 20% of amino acid positions were replaced for substitution augmentations. Augmentation was not applied to PPI training as it distorts the proteins’ structure and properties.

Experimental Setup: All PPI models adopted the same architecture (see Appendix) and were trained for the same number of epochs (50 for MLM pre-training, 15 for PPI training). They were evaluated on the same novel virus test set, with test accuracy and F1 scores as metrics.

3.3. The Agent – Proposing Future More Infectious Virus Variants

Initially, we attempted a Deep Q-Learning Network (DQN) agent (see Appendix). However, it required heavy computation and converged slowly due to our substantial search space of 20L (25,460 actions for SARS-CoV-2). Thus, we explore 2 more efficient algorithms, Greedy Search and Beam Search, to search for the variant with the highest PPI infectivity score. These algorithms rely on the greedy concept: making the optimal choice(s) at each time-step.

```
**Greedy Search Algorithm**
**Inputs:** current sequence S = original spike sequence $S_0$, actions A = 25460
**while** BLOSUM distance <= 40 **do**:
    Perform mutations on S within A to create a batch of variants with shape (25460, 1)
    Tokenize, compute PPI scores and store in array P
    Select the sequence with the highest PPI score, $S_{best} = \text{argmax}(P)$
    Update A = A – 20 actions for mutated position & $S = S_{best}$ if BLOSUM distance with $S_0 \leq 40$
**return S**
```

Algorithm 1. Greedy Search algorithm.

```
**Beam Search Algorithm**
**Inputs:** $S = \{S_0\}$, A = 25460, beamwidth = 10, no. sequences with BLOSUM distance > 40, ($\eta$) = 0
**While** *$\eta$ <beamwidth* **do**:
    **For** *sequence s in S* **do**:
        a. Perform mutations on s to create a batch of variants with shape (25460, 1)
        b. Tokenize, compute PPI scores and store in array P
    Select 10 best sequences with the highest PPI scores, $S_{best} = \text{argmax}_{10}(P)$
    Update S = {for s in $S_{best}$ if BLOSUM distance with $S_0 \leq 40$} and $\eta$ = 10 - length(S)
**return** S
```

Algorithm 2. Beam Search algorithm.

We used Phyre2 [24] to predict the generated variant sequences' 3D structures. Then, possible binding modes of the variants and hACE2 were proposed by the HDOCK server, a hybrid of template-based modeling and free docking which achieved high performance in the Critical Assessment of Prediction of Interaction [25]. We used docking scores as a further metric to validate our proposed variants, where negative scores indicate spontaneous PPIs.

4. Results and Discussion

4.1. PPI Transformer Network Results

Table 1. Performance of the PPI models across all 3 transformer experiments.

No.	MLM	SAM	Data Augmentation	Test Accuracy / %	Loss	F1 Score
1	✓	x	X	44.0	1.080	0.417
2	x	x	x	50.0	4.288	0.658
3	✓	MLM	x	52.0	3.690	0.667
4	✓	MLM + PPI	x	72.0	0.707	0.774
5	✓	PPI	x	74.0	0.642	0.787
6	✓	PPI	Reverse	74.0	0.786	0.787
7	✓	PPI	Dict Sub	78.0	0.910	0.814
8	✓	PPI	Alanine Sub	90.0	0.438	0.906

Experiment 1: Although Model 2 (without MLM pre-training) achieved a higher test accuracy and F1 score than Model 1 (with MLM pre-training), it outputted the same PPI value for all test data (0.99), indicating its inability to learn. In contrast, MLM pre-training helped Model 1 to learn relevant protein features and it outputted different PPI scores for test data. Model 1 was thus used as the baseline for subsequent models to improve upon.

Experiment 2: MLM pre-training with SAM (Models 3 and 4) causes transformer layers that are nearer to the output to learn parameters which improve its predictions of the original AAs being masked (the MLM task). However, these MLM-specific parameters may not be best suited for our PPI task, which uses natural proteins without masking. Thus, SAM on PPI (Model 5) is essential for optimizing the parameters in our PPI task.

Experiment 3: Alanine Sub (Model 8) improved PPI test accuracy the most as it does not drastically alter the protein syntax as compared to Reverse and Dict Sub. This is likely due to alanine’s non polar methyl side-chain (see Appendix), giving rise to alanine’s un reactivity [26]. Model 8 was therefore chosen as the optimal PPI transformer network environment.

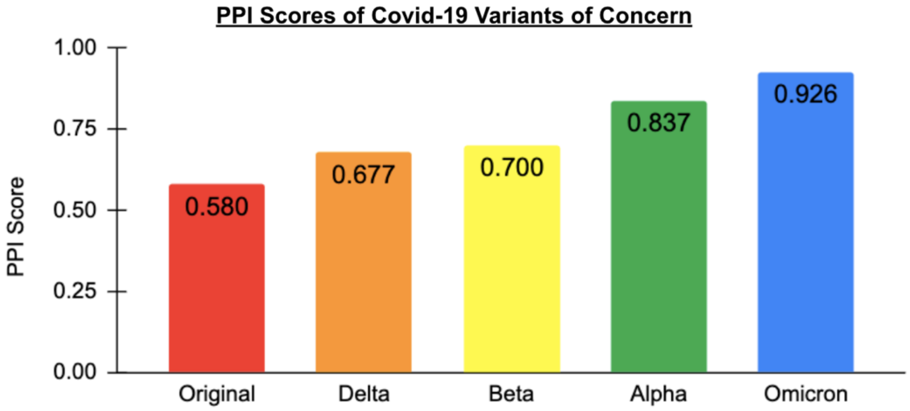


Figure 2. Graph showing PPI scores for different Covid-19 variants of concern.

All variants of concern achieved higher PPI scores than the original spike protein (0.580). The Delta variant (0.677) achieved a higher PPI score than the original, although lower than Alpha (0.837) and Omicron (0.926). These results reflect that our PPI transformer network can make real-world predictions which corroborate well with current Covid-19 research data [27, 28].

4.2. Virus Agent Algorithm Results

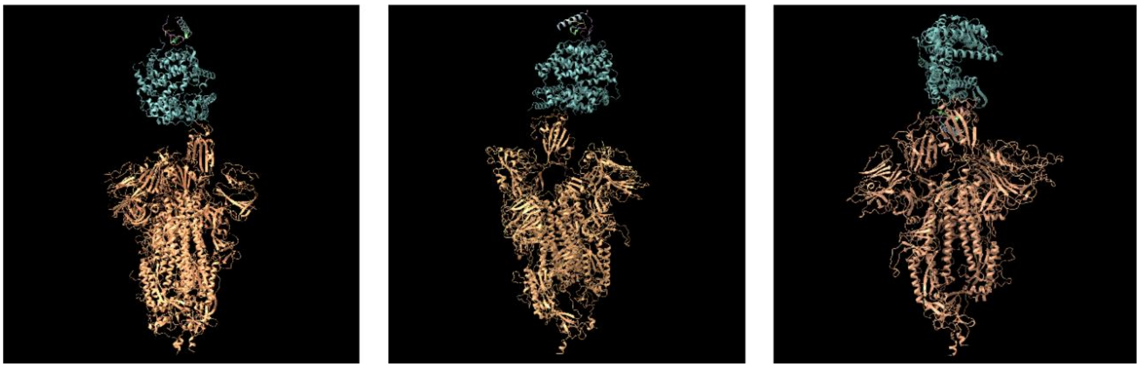


Figure 3. 3D structures of original, greedy and beam search spike proteins bound to hACE2.
Legend: blue: hACE2 receptor, orange: spike glycoprotein.

Table 2. hACE2 binding metrics for the spike glycoprotein variants generated by greedy and beam search.

Algorithm	**PPI Score**	**Docking Score**	**RMSD with S_0 / Å**
Greedy Search	0.99969	–214.76	0.870
Beam Search	0.99973	–218.90	1.095

Based on Table 2, the spike variant proposed by Beam Search attained higher PPI and more negative docking scores than that for Greedy Search, reflecting its greater hACE2 binding affinity. Unlike greedy search which always exploits the best immediate action, beam search strikes a balance between exploitation and exploration (trying a new action). Since AAs in a sequence interact with one another, by considering 10 sequences at each time-step, beam search is likelier to find mutations that may not maximize short-term rewards but will optimize long-term rewards due to future AA interactions. From Figure 3, the proposed variants' structures also have little deviation from the original protein, with RMSDs close to those of current variants (see Appendix). Therefore, an agent armed with a PPI transformer network can propose future more infectious variants, proving our hypothesis. The variants can then be validated experimentally.

5. IMPLICATIONS AND CONCLUSION

We discovered that given an accurate PPI transformer network that measures the infectivity of a proposed variant, our Optimus PPIme approach can effectively predict possible more infectious variants. This narrows the scope of mutated virus proteins for docking and wet-lab testing to validate the variants' infectivity and feasibility.

With only knowledge of the virus and receptor protein sequences, our Optimus PPIme approach can be applied to other dangerous viruses to expedite vaccine development before infectious variants become widespread.

6. LIMITATIONS AND FUTURE WORK

Currently, our Optimus PPIme approach does not consider insertion or deletion mutations in the virus protein, which are also likely to occur in nature. Besides that, the ability of a virus to evade vaccine antibodies is another metric for infectivity, which we did not consider in our Optimus PPIme approach.

Hence, future work can be done to generate insertion or deletion mutations in the virus variant, and to use the evasion of antibodies as a further metric for infectivity.

ACKNOWLEDGEMENTS

We would like to thank our mentor, Dr Shen, for his invaluable guidance and advice throughout this research project!

REFERENCES

[1] World Health Organization. 2021. Tracking SARS-CoV-2 variants. https://www.who.int/en/activities/tracking-SARS-CoV-2-variants/

[2] Barton et al. 2021. Effects of common mutations in the SARS-CoV-2 Spike RBD and its ligand, the human ACE2 receptor on binding affinity and kinetics. Elife. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8480977/

[3] [3] Lovelace, B. 2021. Israel says Pfizer Covid vaccine is just 39% effective as delta spreads, but still prevents severe illness. https://www.cnbc.com/2021/07/23/delta-variant-pfizer-covid-vaccine-39percent-effective-in-israel-prevents-severe-illness.html

[4] [4] Towey, R. 2021. WHO says delta variant accounts for 99% of Covid cases around the world. https://www.cnbc.com/2021/11/16/who-says-delta-variant-accounts-for-99percent-of-covid-cases-around-the-world.html

[5] Sanjuán et al. 2010. Viral Mutation Rates. J Virol: 9733-9748. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2937809/

[6] Zhou, M., Li, Q. and Wang, R. 2016. Current Experimental Methods for Characterizing Protein-Protein Interactions. Wiley Public Health Emergency Collection: 738-756. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7162211/

[7] Zhou et al. 2020. Mutation effect estimation on protein-protein interactions using deep contextualized representation learning. NAR Genomics and Bioinformatics. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7059401/

[8] Xiong, J. 2006. Protein Motifs and Domain Prediction. Essential Bioinformatics: 85-94. https://www.cambridge.org/core/books/abs/essential-bioinformatics/protein-motifs-and-domain-prediction/E17046CB1CD04184A828D8BAC2D222AF

[9] Ofer, D., Brandes, N. and Linial, M. 2021. The language of proteins: NLP, machine learning & protein sequences. Comput Struct Biotechnol J: 1750-1758. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8050421/

[10] Vaswani et al. 2017. Attention Is All You Need. https://arxiv.org/pdf/1706.03762.pdf

[11] Henikoff, S. and Henikoff, J. G. 1992. Amino acid substitution matrices from protein blocks. Proc Natl Acad Sci USA: 10915-10919. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC50453/

[12] Jha et al. 2021. Protein Folding Neural Networks Are Not Robust. https://arxiv.org/pdf/2109.04460.pdf

[13] Stark et al. 2006. BioGRID: a general repository for interaction datasets. Nucleic acids research, 34(Database issue), D535–D539. https://pubmed.ncbi.nlm.nih.gov/16381927/

[14] Guirimand et al. 2015. VirHostNet 2.0: surfing on the web of virus/host molecular interactions data. Nucleic acids research, 43(Database issue), D583–D587. https://pubmed.ncbi.nlm.nih.gov/25392406/

[15] Kshirsagar et al. 2021. Protein sequence models for prediction and comparative analysis of the SARS-CoV-2 −human interactome: Pacific Symposium on Biocomputing, 26, 154–165.https://pubmed.ncbi.nlm.nih.gov/33691013/

[16] Hodcroft, E. 2020. CoVariants. https://covariants.org/

[17] UniProt Consortium. 2021. UniProt: the universal protein knowledgebase in 2021. Nucleic acids research, 49(D1), D480–D489. https://pubmed.ncbi.nlm.nih.gov/33237286/

[18] Lanchantin et al. 2020. Transfer Learning for Predicting Virus-Host Protein Interactions for Novel Virus Sequences. https://doi.org/10.1101/2020.12.14.422772

[19] Devlin et al. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://arxiv.org/abs/1810.04805

[20] Foret et al. 2021. Sharpness-Aware Minimization for Efficiently Improving Generalization. https://arxiv.org/abs/2010.01412

[21] Chen et al. 2021. When Vision Transformers Outperform ResNets without Pre-training or Strong Data Augmentations. https://arxiv.org/abs/2106.01548

[22] Krizhevsky et al. 2012. ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems 25. https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[23] Shen et al. 2021. Improving Generalizability of Protein Sequence Models with Data Augmentations. https://www.biorxiv.org/content/biorxiv/early/2021/02/18/2021.02.18.431877.full.pdf

[24] Kelley et al. 2015. The Phyre2 web portal for protein modelling, prediction and analysis. Nature Protocols: 845-858. https://www.nature.com/articles/nprot.2015.053

[25] Yan et al. 2017. HDOCK: a web server for protein-protein and protein-DNA/RNA docking based on a hybrid strategy. Nucleic Acids Research, Volume 45: 365-373. https://academic.oup.com/nar/article/45/W1/W365/3829194

[26] Lefèvre, F., Rémy, M. H. and Masson, J. M. 1997. Alanine-stretch scanning mutagenesis: a simple and efficient method to probe protein structure and function. Nucleic Acids Research, Volume 25: 447-448. https://academic.oup.com/nar/article/25/2/447/1204328

[27] Mannar et al. 2021. SARS-CoV-2 Omicron Variant: ACE2 Binding, Cryo-EM Structure of Spike ProteinACE2 Complex and Antibody Evasion. https://www.biorxiv.org/content/10.1101/2021.12.19.473380v1.full.pdf

[28] Kim et al. 2021. Differential Interactions Between Human ACE2 and Spike RBD of SARS-CoV-2 Variants of Concern. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8328061/

[29] Rehman, I., Farooq, M. and Botelho, S. 2021. Biochemistry, Secondary Protein Structure. StatPearls [Internet]. https://pubmed.ncbi.nlm.nih.gov/29262225/

AUTHORS

Bingquan Shen is currently a Senior Researcher in DSO National Laboratories. He received his PhD from the National University of Singapore in Mechanical Engineering (Control and Mechatronics) in 2014. His current research interests include limited label learning, adversarial machine learning, and deep learning applications.

Glenda Tan is currently a student at Raffles Institution. Her current research interests include computer vision, transformers and generative adversarial networks.

Koay Tze Erhn is currently a student at Raffles Institution. Her research interests include natural language processing, transformers and computational biology.

APPENDIX

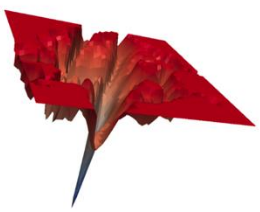
1.1. PPI Dataset Breakdown

Table 3. Breakdown of train and test datasets for the PPI transformer network.

Dataset	Train	Test
BioGRID	11,612 (+)	N/A
VirHostNet	20,257 (+)	26 (–)
Kshirsagar et al.	34,836 (–)	N/A
CoVariants	N/A	24 (+)
Total	66,705	50

In order for our PPI transformer to generalize to unseen virus proteins and make unbiased predictions of future virus variants' infectivity, the training set does not contain V-H interactions involving SARS-CoV-2. Instead, only the PPI transformer's test set contains V-H interactions involving Covid-19 variants and hACE2.

1.2. Our SAM Implementation



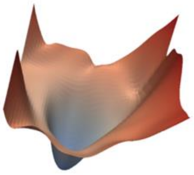


Figure 4. Loss landscapes without SAM training (left) and with SAM training (right)[20].

```
PPI SAM Algorithm
Inputs: neighbourhood ($\rho$) = 0.05, batch size = 8, learning rate ($\alpha$) = 0.001, PPI model's weights $w_0$,
timestep t = 0
While not converged do
    1. Sample batch $B = \{(x_1, y_1), \ldots(x_8, y_8)\}$, where $x = [V_{token}, H_{token}]$
    2. Backpropagation 1: Compute training loss and gradient g
    3. Scale gradient by factor $\rho / (\| g \| + 1e\text{-}12)$ and update weights
    4. Backpropagation 2: Compute training loss and final gradient G
    5. Update weights: $w_{t+1} = w_t - \alpha * G$
    t = t + 1
return $w_t$
```

Algorithm 3. PPI SAM algorithm.

From Figure 4, a loss landscape without SAM training has a sharp global minimum (left) and is difficult to converge, whereas SAM training results in a loss landscape with a wide minimum (right) that is easier to converge [20]. From Algorithm 3, our SAM implementation involves 2 back propagation steps with a scaling step between them.

1.3. Visualization of Protein Augmentation Techniques

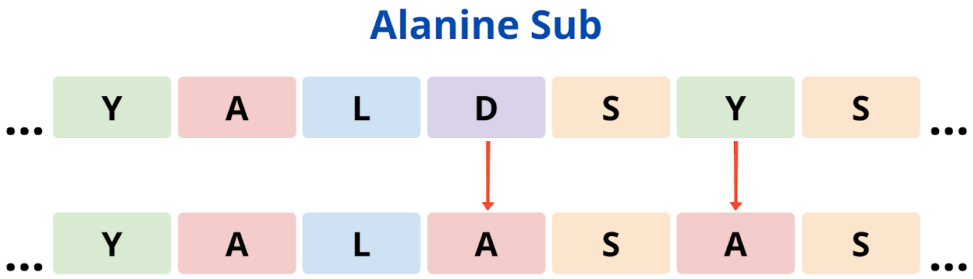


Figure 5a. Visualization of Alanine Sub augmentation.

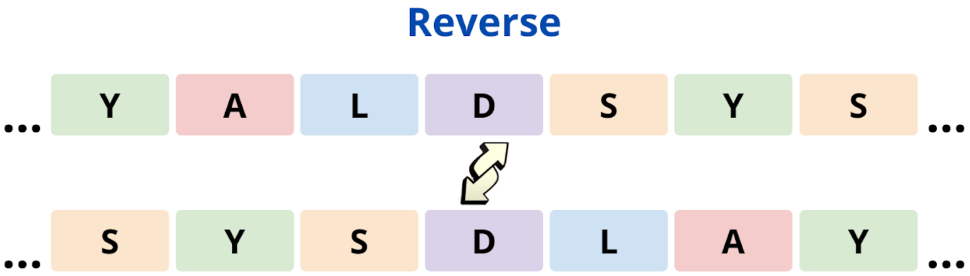


Figure 5b. Visualization of Reverse augmentation.

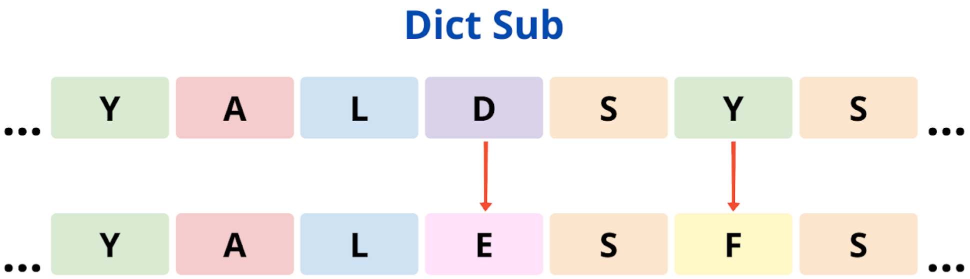


Figure 5c. Visualization of Dict Sub augmentation.

The augmentation techniques were implemented by performing string manipulations on the protein sequence and subsequently feeding the augmented proteins to the transformer network during training. Combinations of augmentations (e.g. all 3, Alanine Sub & Reverse, Alanine Sub & Dictionary Sub, Reverse & Dictionary Sub) were also attempted but they produced models with lower accuracies.

1.4. Skeletal Structure of Amino Acid Alanine

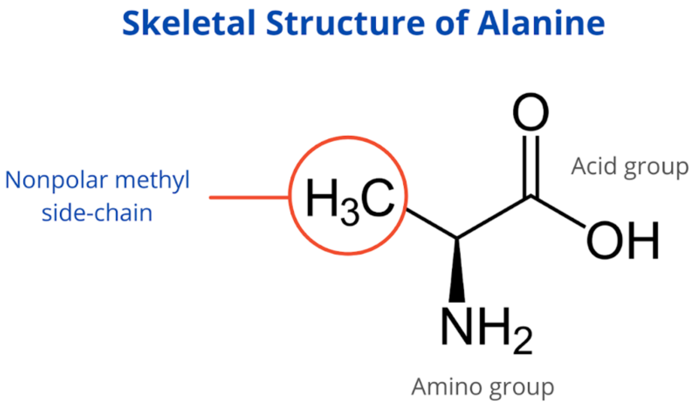


Figure 6. Skeletal structure of alanine with its nonpolar methyl side-chain.

Alanine is a generally unreactive amino acid, attributed to its nonpolar methyl side-chain. This makes Alanine Substitution a suitable candidate for data augmentation.

1.5. Model Architectures

1.5.1. Architecture of the PPI Transformer Network

The PPI transformer network was implemented using the tf.keras framework. The network consists of an encoder to extract features of the virus and receptor proteins, a Concatenate layer to concatenate the encodings of the 2 proteins and finally a 1-unit Dense layer with sigmoid activation which outputs the PPI score. No decoder is used as only an encoder is required for extracting the protein features.

Encoder: The encoder is adapted from the Bidirectional Encoder Representations from Transformers (BERT) encoder and it comprises 3 identical layers. Each layer is made up of a MultiHeadAttention component and a fully-connected feed-forward network. A residual connection is applied around the 2 components, followed by LayerNormalization.

Hyperparameters: After experimenting with different values for the hyperparameters, we arrive at this combination of hyperparameters which produced the best-accuracy model of 90%.

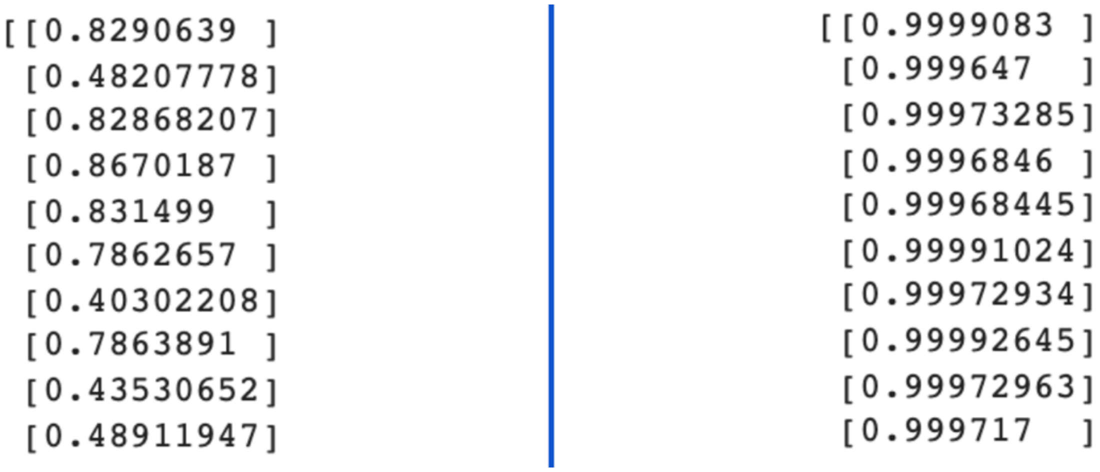
Table 4. Hyperparameters for the PPI transformer encoder.

Hyperparameter	**Value**
Number of transformer layers in encoder	3
Number of heads for MultiHeadAttention	8
Number of embedding dimensions	128
Dropout rate	0.1
Adam Optimizer learning rate	0.001
Batch size	8
Maximum length of proteins	1,300
MLM pre-training epochs	50
PPI training epochs	15

1.5.2. Architecture of the DQN agent

The inefficient DQN agent adopts the same encoder architecture as the PPI transformer network, and the encoder is connected to a 25,460-unit Dense layer with softmax activation. 25,460 represents the number of possible mutations the SARS-CoV-2 spike protein can perform at each time-step (1,273 AA positions x 20 amino acids). The DQN agent was trained for 1000 episodes with a learning rate of 0.7, a batch size of 128 and a minimum replay size of 500.

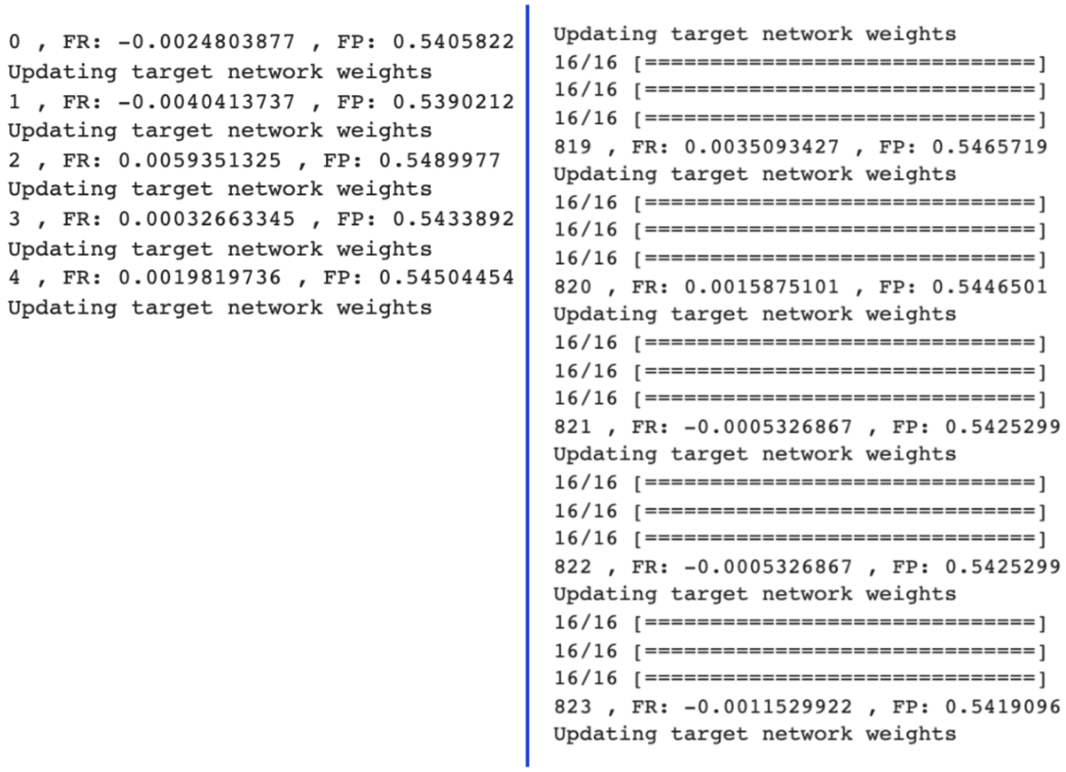
1.6. Failed Results



```
[[0.8290639 ]
 [0.48207778]
 [0.82868207]
 [0.8670187 ]
 [0.831499  ]
 [0.7862657 ]
 [0.40302208]
 [0.7863891 ]
 [0.43530652]
 [0.48911947]
```

```
[[0.9999083 ]
 [0.999647  ]
 [0.99973285]
 [0.9996846 ]
 [0.99968445]
 [0.99991024]
 [0.99972934]
 [0.99992645]
 [0.99972963]
 [0.999717  ]
```

Figure 7. (from left to right) PPI scores outputted by Model 1 (with MLM pre-training) and Model 2 (without MLM pre-training).



```
0 , FR: -0.0024803877 , FP: 0.5405822
Updating target network weights
1 , FR: -0.0040413737 , FP: 0.5390212
Updating target network weights
2 , FR: 0.0059351325 , FP: 0.5489977
Updating target network weights
3 , FR: 0.00032663345 , FP: 0.5433892
Updating target network weights
4 , FR: 0.0019819736 , FP: 0.54504454
Updating target network weights
```

```
Updating target network weights
16/16 [==============================]
16/16 [==============================]
16/16 [==============================]
819 , FR: 0.0035093427 , FP: 0.5465719
Updating target network weights
16/16 [==============================]
16/16 [==============================]
16/16 [==============================]
820 , FR: 0.0015875101 , FP: 0.5446501
Updating target network weights
16/16 [==============================]
16/16 [==============================]
16/16 [==============================]
821 , FR: -0.0005326867 , FP: 0.5425299
Updating target network weights
16/16 [==============================]
16/16 [==============================]
16/16 [==============================]
822 , FR: -0.0005326867 , FP: 0.5425299
Updating target network weights
16/16 [==============================]
16/16 [==============================]
16/16 [==============================]
823 , FR: -0.0011529922 , FP: 0.5419096
Updating target network weights
```

Figure 8. (from left to right) First and last 5 episodes of DQN training.
Legend: FR: final reward (final - initial PPI score of S_0 and hACE2) at the end of each episode, FP: final PPI score at the end of each episode.

From Figure 7, Model 2 outputs the same value of 0.99, indicating its inability to learn. In contrast, MLM pre-training helps Model 1 to output different PPI values, showing that the model has successfully learned the features of proteins. From Figure 8, DQN's final reward does not increase and oscillates between being positive and negative, reflecting the agent's inability to converge.

1.7. Confusion Matrices of PPI Transformer Networks

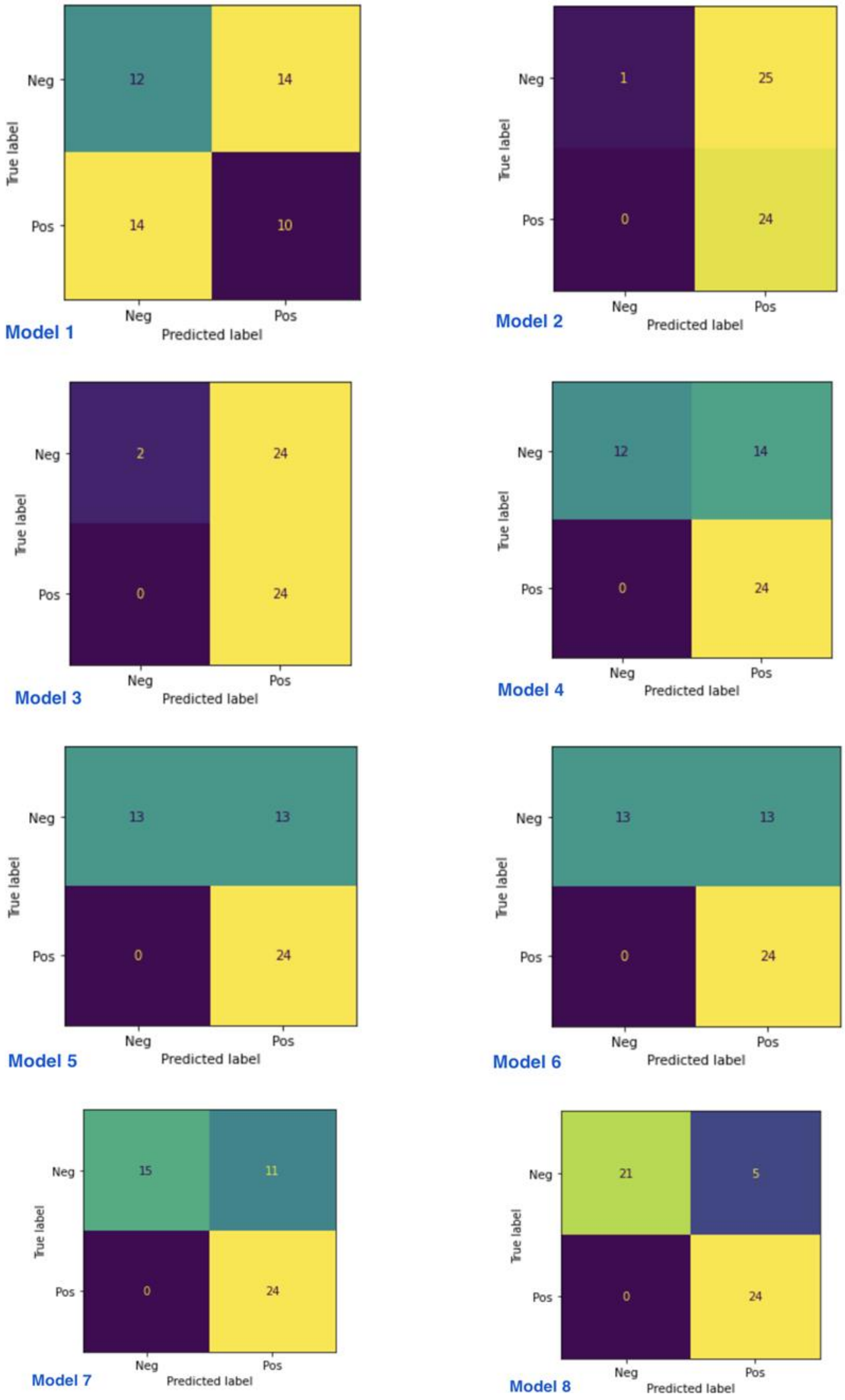


Figure 9. Confusion matrices of PPI transformer networks 1 to 8.

Based on the confusion matrices, Model 8 achieved the highest test accuracy. It correctly predicted all 24 positive interactions and 21 out of 26 negative interactions in our test set.

1.8. Comparison of Our Proposed Variants with Covid-19 Variants of Concern

Table 5: Number of spike mutations and RMSDs of the different variants.

Variant	Number of Spike Mutations	RMSD with S_0 / Å
Alpha	4	0.897
Beta	8	0.901
Delta	8	1.091
Omicron	17	NA
Greedy (Proposed)	6	0.870
Beam (Proposed)	6	1.095

The RMSD values were derived by matchmaking the variants' Protein Data Bank (PDB) files with the original spike protein in ChimeraX. At the time of submission, Omicron's PDB file and hence its RMSD were not yet available due to the variant's recency.

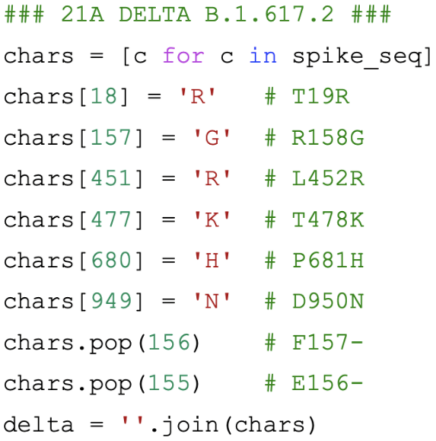
2. ADDITIONAL EXPERIMENTS AND INFORMATION

2.1. More on PPI Dataset Collection

Given a human protein and an unseen novel virus protein, our PPI transformer network must apply what it has learnt from its training data and predict the binding affinity of the new virus-human interaction accurately.

Hence, a PPI in our dataset has to satisfy the following criteria:

a. It is a pair of protein sequences – One being a virus protein and the other being a human host receptor protein.
b. A test PPI comprises the protein of a variant of the target virus and the protein of the human receptor. For our current experiments on SARS-CoV-2, the target virus protein is the SARS-CoV-2 spike glycoprotein while the human protein is the hACE2 receptor.
c. A training PPI cannot include proteins of the target virus and its variants. If the transformer network is trained on existing sequences of the target virus, this would distort the model's learning parameters and cause it to make biased predictions on a new variant-host interaction. Hence, for our current experiments on SARS-CoV-2, we train our transformer network on PPIs excluding the proteins of SARS-CoV-2 and its variants.



```
### 21A DELTA B.1.617.2 ###
chars = [c for c in spike_seq]
chars[18] = 'R'    # T19R
chars[157] = 'G'   # R158G
chars[451] = 'R'   # L452R
chars[477] = 'K'   # T478K
chars[680] = 'H'   # P681H
chars[949] = 'N'   # D950N
chars.pop(156)     # F157-
chars.pop(155)     # E156-
delta = ''.join(chars)
```

Figure 10. Generation of the Delta variant's spike glycoprotein sequence from listed mutations.

Due to the recency of various Covid-19 variants of concern, there were no accessible protein databases with the sequences of the variants' spike glycoproteins. Thus, to obtain the spike glycoprotein sequences of the Covid-19 variants for the novel virus test set, we identified 23 variants listed in CoVariants [16] and modified the original SARS-CoV-2 spike glycoprotein's sequence following the identified mutations in CoVariants, producing the variants' sequences.

Using the Delta variant in Figure 10 as an example, we first split the spike protein sequence of the original SARS-Cov-2 virus into a list of its individual characters. For each substitution mutation, we modify the amino acid at the given position and replace it with the new given amino acid. For example, if the mutation is T19R, we replace amino acid T at position 19 with amino acid R. If the mutation is an insertion mutation, we insert an amino acid character into the list. If the mutation is a deletion mutation, we pop the amino acid at the given position. For the mutation F157-, we remove amino acid F at position 157.

Our compiled spike glycoprotein sequences of the SARS-CoV-2 virus and its 23 variants can be found at this link: https://tinyurl.com/covidvariantsequences

2.2. Additional Experiment: Secondary Structure Prediction Pre-training

2.2.1. Introduction to Secondary Structure Prediction

Initially, our transformer predicted the same value of 0.99 for all test data (see Appendix), and this was resolved with the addition of MLM pre-training. Besides pre-training our transformer on MLM, we attempted secondary structure prediction as another pre-training technique. This could enable our transformer to learn more structural features of proteins, which also play an important role in determining whether two proteins interact.

Unlike the primary structure of proteins (a sequence of amino acids), the secondary structure of proteins comprises regions stabilized by hydrogen bonds between atoms in the protein backbone, giving rise to the regular, local structure of the protein backbone. Common secondary structures are the alpha-helix (the shape of a clockwise spiral staircase), beta-pleated sheet (2 sections of protein chain aligned side-by-side in an extended conformation), and loops (relatively disordered protein segments) [29].

2.2.2. Training Details and Tests Performed

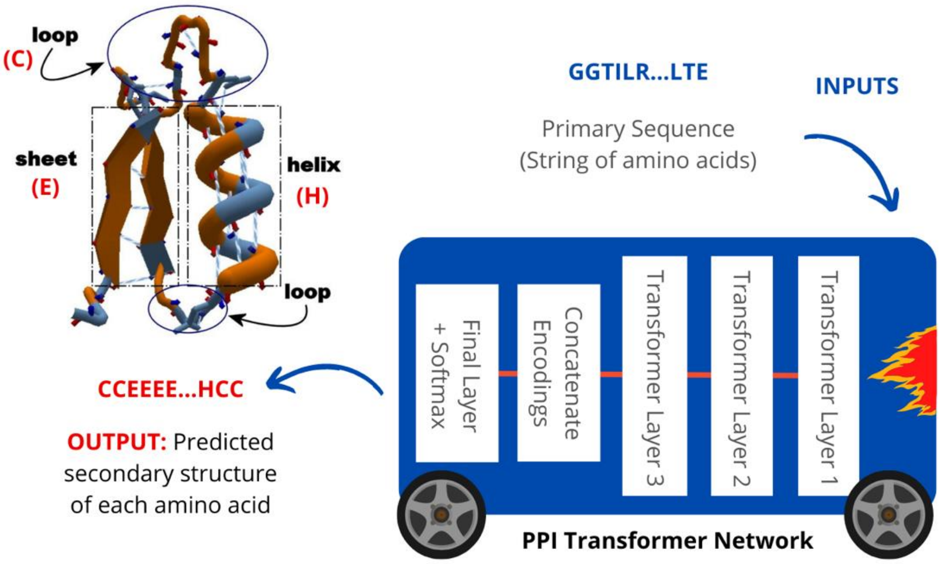


Figure 11. Overview of the secondary structure prediction pre-training.

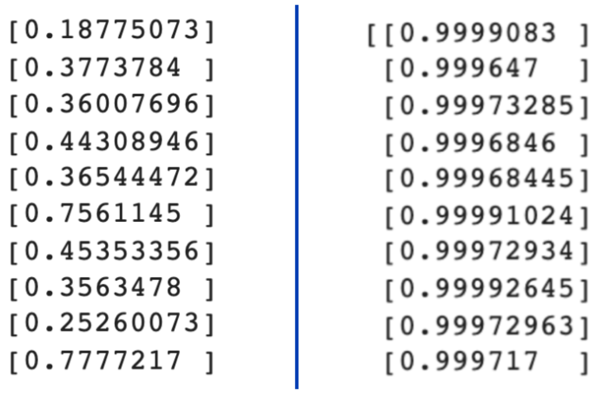
For the secondary structure prediction task, as there was no secondary structure data for the proteins in our current PPI dataset, we pre-trained our transformer on secondary structure prediction datasets from Kaggle. The training data consists of a sequence of amino acids of length L, while the label is a tagged sequence of the same length, where each position denotes the secondary structure class that each amino acid belongs to (H for alpha-helix, E for beta-sheet, and C for loop).

Below were the tests performed to determine the effectiveness of secondary structure prediction.

a. *Model 9*: Pre-training the transformer network on only secondary structure prediction, followed by PPI training.
b. *Model 10*: Pre-training the transformer network on both secondary structure prediction and MLM, followed by PPI training.

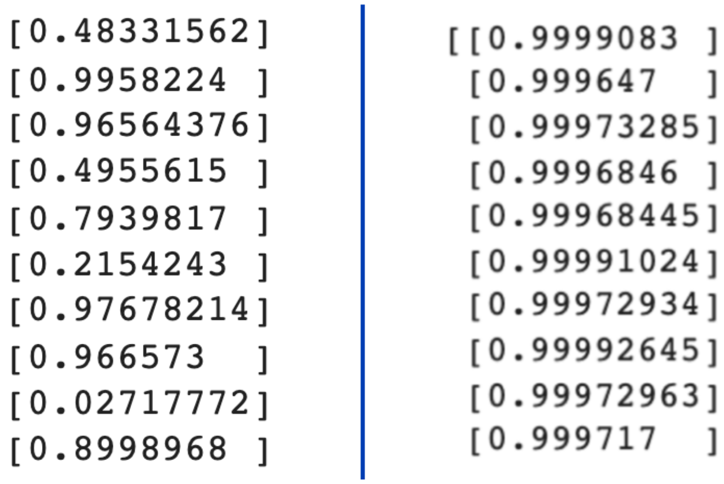
Experimental Setup: The transformer network adopts the same encoder architecture and hyperparameters as those in Section 10.5.1., however the final layer uses softmax activation rather than sigmoid activation. The transformer network was pre-trained on secondary structure prediction for 25 epochs, MLM for 50 epochs and trained on the PPI task for 15 epochs before being evaluated on the same novel virus test set.

2.2.3. Results of the Secondary Structure Prediction Tests



```
[0.18775073]        [[0.9999083 ]
[0.3773784 ]         [0.999647  ]
[0.36007696]         [0.99973285]
[0.44308946]         [0.9996846 ]
[0.36544472]         [0.99968445]
[0.7561145 ]         [0.99991024]
[0.45353356]         [0.99972934]
[0.3563478 ]         [0.99992645]
[0.25260073]         [0.99972963]
[0.7777217 ]         [0.999717  ]
```

Figure 12. (from left to right) PPI scores outputted by Model 9 (with secondary structure prediction pre-training only) and Model 2 (no pre-training).



```
[0.48331562]        [[0.9999083 ]
[0.9958224 ]         [0.999647  ]
[0.96564376]         [0.99973285]
[0.4955615 ]         [0.9996846 ]
[0.7939817 ]         [0.99968445]
[0.2154243 ]         [0.99991024]
[0.97678214]         [0.99972934]
[0.966573  ]         [0.99992645]
[0.02717772]         [0.99972963]
[0.8998968 ]         [0.999717  ]
```

Figure 13. (from left to right) PPI scores outputted by Model 10 (with both MLM and secondary structure prediction pre-training) and Model 2 (no pre-training).

From Figure 12, implementing secondary structure prediction pre-training helped Model 9 to learn structural features of proteins, enabling it to output different PPI scores. From Figure 13, combining both MLM and secondary structure prediction pre-training likewise helped Model 10 to output different PPI scores. This proves that secondary structure prediction is an effective pre-training strategy. In future, SAM training and data augmentation can be applied to models pre-trained on secondary structure prediction so that the effectiveness of this pre-training technique can be fully compared with that of MLM pre-training.

Evaluation of secondary structure prediction and MLM pre-training: MLM pre-training would be the winner in terms of the ease of implementation as it is self-supervised and leverages the primary sequences of existing training proteins in the dataset. Contrastingly, secondary structure prediction is a supervised learning technique which requires the collection of additional labelled data, and not all the proteins in the PPI dataset have data on their secondary structures. Hence, implementing MLM pre-training solely can produce robust PPI models with high accuracies and it is usually sufficient.

2.3. More on Protein Data Augmentation

2.3.1. Selection of Protein Data Augmentation Techniques

There are other protein data augmentation techniques besides Alanine Sub, Dict Sub and Reverse. One such technique is Shuffling, which randomly shuffles the amino acid sequence. Another technique is Sampling, which retains a selected local region of the amino acid sequence and removes unselected amino acid positions [23].

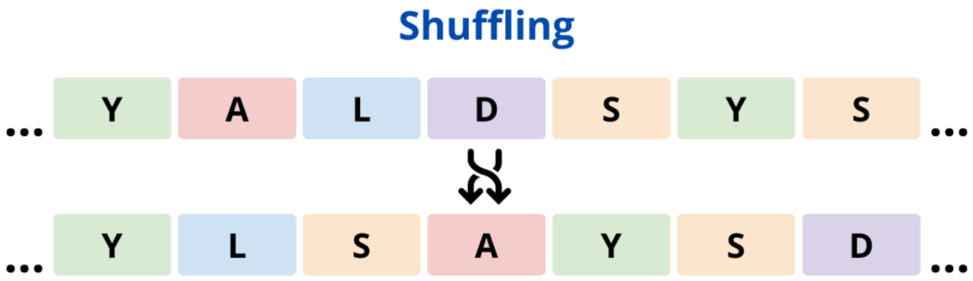


Figure 14a. Visualization of Shuffling augmentation.

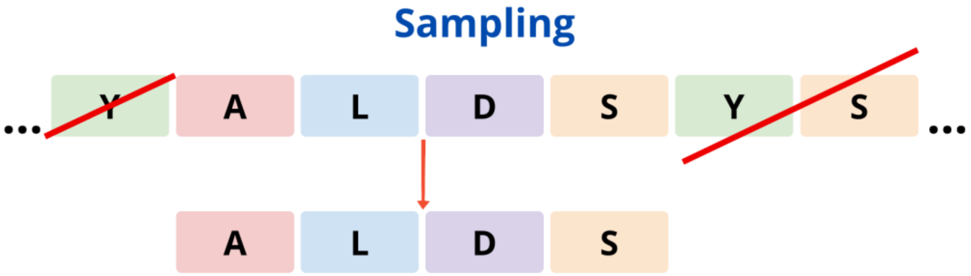


Figure 14b. Visualization of Sampling augmentation.

However, Shuffling and Sampling were not used for our experiments as they alter the protein syntax too drastically. For Shuffling, the whole sequence of amino acids is destroyed, which causes the augmented protein's structural and chemical properties to be entirely different, since a protein's primary sequence gives rise to these properties. For Sampling, numerous amino acid positions are removed, so the augmented protein is incomplete and removed amino acids that also participate in the PPI are not seen by the model. Hence, these augmentation techniques alter the features of the protein sequences to the extent that they lower the effectiveness of MLM pre-training, since most features of the original proteins are no longer captured. Alanine Sub, Dict Sub and Reverse were used instead since the augmented protein has the same length and a similar sequence to the original protein, thus they will have similar structural and chemical properties.

2.3.2. Additional Experiment: Combining Protein Data Augmentation Techniques

Given that implementing Alanine Sub, Dict Sub and Reverse individually could improve our PPI test accuracies in Experiment 3, we aim to determine if combining the augmentation techniques could further improve results. The PPI transformer network was trained on both Alanine Sub and

Reverse (Model 11). Dict Sub was not included as it produces similar effects to Alanine Sub and Alanine Sub was proven to be more effective from Experiment 3.

Experimental Setup: Model 11 adopted the same architecture as that in Experiment 3 and it was trained for the same number of epochs (50 for MLM pre-training, 15 for PPI training) before evaluation on the same novel virus test set. Like in Experiment 3, 25% of proteins were augmented and 20% of amino acid positions were replaced for Alanine Sub. Augmentation was not applied to PPI training as it distorts the proteins' structure and properties.

Table 6. Performance of the PPI models trained on different augmentation techniques.

No.	MLM	SAM	Data Augmentation	Test Accuracy / %	Loss	F1 Score
6	✓	PPI	Reverse	74.0	0.786	0.787
7	✓	PPI	Dict Sub	78.0	0.910	0.814
8	✓	**PPI**	**Alanine Sub**	**90.0**	**0.438**	**0.906**
11	✓	PPI	Alanine Sub + Reverse	78.8	0.473	0.728

Based on Table 6, Model 11 (Alanine Sub + Reverse) scored a higher test accuracy than Model 6 (Reverse only) and Model 7 (Dict Sub only), but it performed worse than Model 8 (Alanine Sub only) as shown by the lower test accuracy, lower F1 score and higher loss. Since Reverse is less effective than Alanine Sub in improving PPI test accuracies, the addition of Reverse may have hurt the effectiveness of Alanine Sub and rendered it unable to achieve its full performance. Thus, applying Alanine Sub solely to the PPI transformer still yields the best results.

2.3.3. Additional Experiment: Varying the Percentage of Augmented Proteins

The experiments so far have shown that Alanine Sub is the most effective augmentation technique, improving PPI test accuracy to 90%. However, there are hyperparameters that have yet to be explored, such as the percentage of augmented proteins in the training set. Hence, we pre-trained PPI models on different values of this hyperparameter for Alanine Sub. One model was trained with 20% of proteins augmented (Model 12), while another was trained with 33% of proteins augmented (Model 13).

Experimental Setup: The models adopted the same architecture as that in Experiment 3 and were trained for the same number of epochs (50 for MLM pre-training, 15 for PPI training) before evaluation on the same novel virus test set. Like in Experiment 3, 20% of amino acid positions were replaced for Alanine Sub. Augmentation was not applied to PPI training as it distorts the proteins' structure and properties.

Table 7. Performance of Alanine Sub PPI models trained on different augmentation rates.

No.	Proteins augmented / %	MLM	SAM	Test Accuracy / %	Loss	F1 Score
8	**25**	✓	**PPI**	**90.0**	**0.438**	**0.906**
12	20	✓	PPI	85.0	0.362	0.851
13	33	✓	PPI	86.3	0.400	0.851

From Table 7, Model 12 (20% of proteins were augmented) scored the lowest accuracy, followed by Model 13 (33% of proteins were augmented) and Model 8 (25% of proteins were augmented) still scored the highest accuracy. This shows that initially increasing the percentage of augmented proteins increases performance as there is more training data available for the transformer network to optimize its weights on. However, increasing the percentage of augmented proteins beyond 25% causes performance to drop. Thus, 25% is likely the optimal hyper parameter value.